SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

INSTITUTE OF PLACEMENT AND TRAINING

CSA09 –JAVA PROGRAMMING

String

1. Write a program to reverse a word using loop? (Not to use inbuilt functions)
   Sample Input:
   String: TEMPLE
   Sample Output:
   Reverse String: ELPMET

```
import java.io.*;
import java.util.*;
class name
{
public static void main(String args[])
{
String input="TEMPLE";
String reversed="";
for (int i=input.length()-1;i>=0 ;i--)
{
reversed += input.charAt(i);
}
System.out.println("reversed string:"+reversed);
}
}
```

2. Write a program to convent the given string to integer?
   Sample Input:
   String: 1234
   Sample Output:
   Out put String: 1234

3. Write a program to check the entered user name is valid or not. Get both the inputs from the user.

```
import java.io.*;
import java.util.*;
class word
{
public static void main(String args[])
{
String s1,s2;
boolean result;
Scanner s = new Scanner(System.in);
System.out.println("enter the string 1:");
s1 = s.nextLine();
System.out.println("enter the string 2:");
s2 = s.nextLine();
result = s1.equals(s2);
if (result == false)
{
System.out.println("username is invalid");
}
else
{
System.out.println("username is valid");
}
}
}
```

4. Write a program that would sort a list of names in alphabetical order Ascending or Descending, choice get from the user?

Sample Input:

```
    Banana
    Carrot
    Radish
    Apple
    Jack
    Order(A/D) : A
    Sample Output:
    Apple
    Banana
    Carrot
    Jack
    Radish
```

5. Write a program to print the special characters separately and print number of Special characters in the line?

```
import java.util.Scanner;

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

        // Input a line
        System.out.print("Enter a line of text: ");
        String inputLine = scanner.nextLine();

        // Count and print special characters
        int specialCharCount = 0;
        System.out.println("Special characters in the line:");

        for (char c : inputLine.toCharArray()) {
            if (isSpecialCharacter(c)) {
                System.out.print(c + " ");
                specialCharCount++;
            }
        }

        // Print the total count of special characters
        System.out.println("\nNumber of special characters: " + specialCharCount);
    }

    private static boolean isSpecialCharacter(char c) {
        // Check if the character is a special character
        return !Character.isLetterOrDigit(c) && !Character.isWhitespace(c);
    }
}
```

6. Write a program to print the number of vowels in the given statement?

Sample Input:

```
    Saveetha School of Engineering
```

Sample Output:

```
    Number o vowels = 12
```

```
import java.util.*;

public class vowellss {
    public static void main(String[] Args){
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a Stirng : ");
        String input = sc.nextLine();
        int count = 0;

        for(int i = 0;i<input.length();i++){
            if((input.charAt(i) == 'A' )||(input.charAt(i) == 'E' ) ||(input.charAt(i) == 'I'
)||(input.charAt(i) == 'O' )||(input.charAt(i) == 'U' )||(input.charAt(i) == 'a' )||(input.charAt(i)
== 'e' )||(input.charAt(i) == 'i' )||(input.charAt(i) == 'o' )||(input.charAt(i) == 'u' )){
                count +=1;
            }
        }
        System.out.println("No. of Vowels : " + count );

    }
}
```

7. Write a program to print consonants and vowels separately in the given word

   Sample Input:

   Given Word: Engineering

   Sample Output:

   Consonants: n g n r n g

   Vowels: e i e ei

```
import java.io.*;
import java.util.*;
class name
{
public static void main(String args[])
{
String input;
System.out.println("enter the string:");
Scanner s = new Scanner(System.in);
input = s.nextLine();
String vow="";
String con="";
for (int i=0;i<input.length();i++)
{
if((input.charAt(i)=='A')||(input.charAt(i)=='E')||(input.charAt(i)=='I')||(input.charAt(i)=='O')||(input.charAt(i)=='a')||(input.charAt(i)=='e')||(input.charAt(i)=='i')||(input.charAt(i)=='i')||(input.charAt(i)=='o')||(input.charAt(i)=='u'))
{
vow = vow + input.charAt(i);
}
else
{
con = con + input.charAt(i);
}
}
System.out.println("vowels:"+vow);
System.out.println("consonants:"+con);
}
}
```

8. Write a program that finds whether a given character is present in a string or not. In case it is present it prints the index at which it is present. Do not use built-in find functions to search the character.

   Sample Input:

   Enter the string: I am a programmer

   Enter the character to be searched: p

   Sample Output:

   P is found in string at index: 8

   Note: Check for non available Character in the given statement as Hidden Test case.

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

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

        // Input a string
        System.out.print("Enter the string: ");
        String inputString = scanner.nextLine();

        // Input the character to be searched
        System.out.print("Enter the character to be searched: ");
        char searchChar = scanner.next().charAt(0);

        // Search for the character in the string
        int index = findCharacterIndex(inputString, searchChar);

        // Display the result
        if (index != -1) {
            System.out.println(searchChar + " is found in string at index: " + index);
        } else {
            System.out.println(searchChar + " is not found in the string.");
        }
    }

    private static int findCharacterIndex(String str, char targetChar) {
        // Iterate through the string to find the character
        for (int i = 0; i < str.length(); i++) {
            if (str.charAt(i) == targetChar) {
                return i; // Return the index if the character is found
            }
        }
        return -1; // Return -1 if the character is not found
    }
}
```

9. Write a program to arrange the letters of the word alphabetically in reverse order

Sample Input:

Enter the word: MOSQUE

Sample Output:

Alphabetical Order: U S Q O M E

Test Case:

1. HYPOTHECATION
2. MATRICULATION
3. MANIPULATION

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

```
class name
{
public static void main(String args[])
{
String s;
System.out.println("enter the string:");
Scanner p = new Scanner(System.in);
s = p.nextLine();

char arr[] = s.toCharArray();
char temp;
int i=0;
while (i<arr.length)
{
int j=i+1;
while (j<arr.length)
{
if (arr[j] > arr[i])
{
temp = arr[i];
arr[i] = arr[j];
arr[j] = temp;
}
j += 1;
}
i += 1;
}
System.out.println(arr);
}
}
```

10. Write a program that accepts a string from user and displays the same string after removing vowels from it.

Sample Input & Output:

```
Enter a string: we can play the game
The string without vowels is: w cn ply thgm
```

```
import java.io.*;
import java.util.*;
class name
{
public static void main(String args[])
{
String s1,s2;
Scanner p = new Scanner(System.in);
System.out.println("enter a string:");
s1 = p.nextLine();
s2 = s1.replaceAll("[aeiouAEIOU]","");
System.out.println(s2);
}
}
```

Arrays:

11. Write a program for matrix multiplication?

```
Sample Input:
Mat1 =      1  2
            5  3
Mat2 =      2   3
            4   1
Sample Output:
Mat Sum = 10    5
           22    18
```

```
import java.io.*;
import java.util.*;
class name
{
public static void main(String args[])
{
int a[][] = {{1,2},{3,4}};
int b[][] = {{5,6},{7,8}};
int c[][] = new int[2][2];
for (int i=0;i<2;i++)
{
for (int j=0;j<2;j++)
{
c[i][j] = a[i][j] * b[i][j];
System.out.println(c[i][j]+"");
}
}
System.out.println();
}
}
```

12. Write a program for matrix addition?

```
Sample Input:
Mat1 =      1  2
            5  3
Mat2 =      2   3
            4   1
Sample Output:
Mat Sum = 3    5
            9     4
```

```
import java.io.*;
import java.util.*;
class name
{
public static void main(String args[])
{
int a[][] ={{1,2},{3,4}};
int b[][] ={{5,6},{7,8}};
int c[][] = new int[2][2];
for (int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
c[i][j] = a[i][j] + b[i][j];
System.out.println(c[i][j]+" ");
}
System.out.println();
}
}
}
```

13. Write a program for Merge two sorted arrays using Array list
Input: arr1[] = { 1, 3, 4, 5}, arr2[] = {2, 4, 6, 8}
Output: arr3[] = {1, 2, 3, 4, 4, 5, 6, 8}

14. Find the Mean, Median, Mode of the array of numbers?
Sample Input;:
Array of elements = {16, 18, 27, 16, 23, 21, 19}
Sample Output:
Mean = 20
Median = 19
Mode = 16
Test cases:
1. Array of elements = {26, 28, 37, 26, 33, 31, 29}
2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 160, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100, 100}

15. Write a program to find the number of composite numbers in an array of elements
Sample Input;:
Array of elements = {16, 18, 27, 16, 23, 21, 19}
Sample Output:
Number of Composite Numbers = 5
Test cases:
1. Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int x[] = new int[7];
Scanner s = new Scanner(System.in);
System.out.println("enter the elements:");
int count=0;
for(int i=0;i<7;i++)
{
x[i] = s.nextInt();
}
for(int i=0;i<7;i++)
{
if(isComposite(x[i]))
{
count++;
}
}
System.out.println("composite numbers are:"+count);
}
private static boolean isComposite(int n)
{
if(n<=1)
{
return false;
}
for (int i=2;i<=Math.sqrt(n);i++)
{
if(n % i == 0)
{
return true;
}
}
return false;
}
}
```

Patterns :

16. Write a program to print Right Triangle Star Pattern

Sample Input:: n = 5

Output:

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int n;
Scanner s = new Scanner(System.in);
System.out.println("enter n value:");
n = s.nextInt();
for(int i=1;i<=n;i++)
{
for(int j=1;j<=n-i;j++)
{
System.out.print(" ");
}
for(int k=1;k<=i;k++)
{
System.out.print("* ");
}
System.out.println();
}
}
}
```

17. Write a program to print the below pattern?

```
                1

            1       1

        1       2       1

    1       3       3       1

1       4       6       4       1
```

import java.io.*;
import java.util.*;
class rec

```
{
public static void main(String args[])
{
int rows;
Scanner s = new Scanner(System.in);
rows = s.nextInt();
for(int i=0;i<=rows;i++)
{
for(int j=0;j<=rows-i;j++)
{
System.out.print(" ");
}
int value=1;
for(int j=0;j<=i;j++)
{
System.out.print(value+"  ");
value = value * (i-j)/(j+1);
}
System.out.println();
}
}
}
```

18. Write a program to print rectangle symbol pattern.
Get the symbol as input from user

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int rows,colums;
char symbol;
Scanner s = new Scanner(System.in);
System.out.println("enter no.of rows:");
rows = s.nextInt();
System.out.println("enter no.of columns:");
colums = s.nextInt();
System.out.println("enter the symbol:");
symbol = s.next().charAt(0);
for(int i=0;i<=rows;i++)
{
for(int j=0;j<=colums;j++)
{
System.out.print(symbol+" ");
}
System.out.println();
}
}
}
```

19. Write a program to print the following pattern

Sample Input:

```
Enter the number to be printed: 1
Max Number of time printed: 3
1
11
111
11
1
```

```
import java.util.Scanner;

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

        // Get input from the user
        System.out.print("Enter the number to be printed: ");
        int numToPrint = scanner.nextInt();

        System.out.print("Max Number of times printed: ");
        int maxPrint = scanner.nextInt();

        // Call the method to print the pattern
        printPattern(numToPrint, maxPrint);
    }

    // Method to print the pattern
    static void printPattern(int num, int maxTimes) {
        // Print the upper half of the pattern
        for (int i = 1; i <= maxTimes; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(num);
            }
            System.out.println();
        }

        // Print the lower half of the pattern
        for (int i = maxTimes - 1; i > 0; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print(num);
            }
            System.out.println();
        }
    }
}
```

20. Write a program to print the Inverted Full Pyramid pattern?

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int rows;
Scanner s = new Scanner(System.in);
System.out.println("enter no.of rows:");
rows = s.nextInt();
for(int i=rows;i>=1;i--)
{
for(int j=0;j<rows-i;j++)
{
System.out.print("   ");
}
for(int j=1;j<=2*i-1;j++)
{
System.out.print(" * ");
}
System.out.println();
}
}
}
```

21. Write a program to print the following pattern

Sample Input:

Enter the Character to be printed: %

Max Number of time printed: 3

```
%
% %
% % %
```

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
```

```
int maxtimes;
char symbol;
Scanner s = new Scanner(System.in);
System.out.println("enter max times to be printed:");
maxtimes = s.nextInt();
System.out.println("enter the character to print:");
symbol = s.next().charAt(0);
for(int i=1;i<=maxtimes;i++)
{
for(int j=1;j<=i;j++)
{
System.out.print(symbol+ " ");
}
System.out.println();
}
}
}
```

22. Write a program to print hollow square symbol pattern?

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int sidelength;
char symbol;
Scanner s = new Scanner(System.in);
System.out.println("enter the sidelength:");
sidelength  = s.nextInt();
System.out.println("enter the symbol:");
symbol = s.next().charAt(0);
for(int i=1;i<=sidelength;i++)
{
for(int j=1;j<=sidelength;j++)
{
if(i==1 || i==sidelength || j==1 || j==sidelength)
{
System.out.print(symbol + " ");
}
else
{
System.out.print("  ");
}
}
System.out.println();
}
}
}
```

23. Write a program to print the below pattern

```
1
2  2
3  3  3
4  4   4   4
```

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int rows;
Scanner s = new Scanner(System.in);
System.out.println("enter no.of rows:");
rows = s.nextInt();
for(int i=1;i<=rows;i++)
{
for(int j=1;j<=i;j++)
{
System.out.print(i + " ");
}
System.out.println();
}
}
}
```

24. Write a program to print the below pattern

```
1
4     9
16   25    36
49   64    81    100
```

25. Write a program to print the below pattern

```
1
2 2
3  3  3
4  4   4   4
3  3  3
2  2
1
```

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

```
class rec
{
public static void main(String args[])
{
int rows;
Scanner s = new Scanner(System.in);
System.out.println("enter no.of rows:");
rows = s.nextInt();
for(int i=1;i<=rows;i++)
{
for(int j=0;j<=i;j++)
{
System.out.print(i+" ");
}
System.out.println();
}
for(int i=rows-1;i>=1;i--)
{
for(int j=1;j<=i;j++)
{
System.out.print(i + " ");
}
System.out.println();
}
}
}
```

26. Write a program to print hollow Square Dollar pattern?

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int sidelength;
char symbol;
Scanner s = new Scanner(System.in);
System.out.println("enter the sidelength:");
sidelength  = s.nextInt();
System.out.println("enter the symbol:");
symbol = s.next().charAt(0);
for(int i=1;i<=sidelength;i++)
{
for(int j=1;j<=sidelength;j++)
{
if(i==1 || i==sidelength || j==1 || j==sidelength)
{
System.out.print(symbol + " ");
}
else
{
System.out.print("  ");
}
}
System.out.println();
}
}
}
```

27. Write a program to print inverted pyramid pattern.
Input: no of rows: 3
Output

```
*****
 ***
  *
```

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int rows;
Scanner s = new Scanner(System.in);
System.out.println("enter no.of rows:");
rows = s.nextInt();
for(int i=rows;i>=1;i--)
{
for(int j=0;j<rows-i;j++)
{
System.out.print("   ");
}
for(int j=1;j<=2*i-1;j++)
{
System.out.print(" * ");
}
System.out.println();
}
}
}
```

General:

28. Write a program to reverse a number using loop?(Get the input from user)

Sample Input:

Number: 14567

Sample Output:

Reverse Number: 76541

Test cases:

1. -45721
2. 000
3. AD1947
4. !@#$%
5. 145*999=144855

```
import java.io.*;
import java.util.*;
class program
{
public static void main(String arg[])
{
try
{
Scanner sc=new Scanner(System.in);
int n,re=0,rem;
System.out.println("Enter a number:");
n=sc.nextInt();
while(n!=0)
{
  rem=n%10;
  re=re*10+rem;
  n=n/10;
}
System.out.println("The reversed number is:"+re);
}
catch(Exception e)
{
System.out.println("Enter a valid number");
}
}
}
```

29. Write a program to convert the given decimal to binary and print the reverse of the binary decimal.

Input: 11

Output: 13

Explanation: (11)10 = (1011)2.

After reversing the bits we get:

(1101)2 = (13)10.

Test cases:

1. 25
2. Eighteen
3. 12
4. -18
5. 34.5

30. Write a program to find whether the person is eligible for vote or not. And if that particular person is not eligible, then print how many years are left to be eligible.

Sample Input:

```
Enter your age:          7
```

Sample output:

```
You are allowed to vote after 11 years
```

Test cases:

6. 25
7. Eighteen
8. 12
9. -18
10. 34.5

```
import java.io.*;
import java.util.*;
class poojaa
{
public static void main(String args[])
{
try
{
int age;
System.out.println("enter the age of person:");
Scanner s = new Scanner(System.in);
age = s.nextInt();
if(age>=18)
{
System.out.println("eligible to vote");
}
else
{
int years = 18 - age;
System.out.println("you are eligible after"+years+"years");
}
}
catch (Exception e)
{
System.out.println("enter the valid age");
}
}
}
```

31. Find the LCM and GCD of n numbers?

Sample Input:

N value = 2
Number 1 = 16
Number 2 = 20

Sample Output:

LCM = 80
GCD = 4

Test cases:

1. N = 3, {12, 25, 30}
2. N = 2, {52, 25, 63}
3. N = 3, {17, 19, 11}
4. N = -2, {52, 60}
5. N = 2, {30, 45}

```
import java.util.Scanner;
import java.io.*;

class sree
{
static int gcd(int x, int y)
{
 int r=0, a, b;
 a = (x > y) ? x : y;
 b = (x < y) ? x : y;
 r = b;
 while(a % b != 0)
 {
  r = a % b;
  a = b;
  b = r;
 }
 return r;
}
static int lcm(int x, int y)
{
 int a;
 a = (x > y) ? x : y;
 while(true)
 {
  if(a % x == 0 && a % y == 0)
      return a;
       ++a;
 }
 }
public static void main(String args[])
 {
 try
 {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter N value: ");
        int N=sc.nextInt();
        if(N==2)
        {

        System.out.println("Enter the two numbers: ");
        int x = sc.nextInt();
        int y = sc.nextInt();

        System.out.println("The GCD of two numbers is: " + gcd(x, y));
        System.out.println("The LCM of two numbers is: " + lcm(x, y));
  }
     }
      catch(Exception e)
        {
           System.out.println("Enter only numbers");

        }
     }
}
```

32. Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. He is being offered 12 percent rate of interest; for all other customers, the ROI is 10 percent.

Sample Input:

Enter the principal amount: 200000
Enter the no of years: 3
Is customer senior citizen (y/n): n

Sample Output:

Interest: 60000

Test Cases:

1. Principal: 2000 , Years: 0
2. Principal: 20000 , Years: -2
3. Principal: -2000 , Years: 2
4. Principal: 2 , Years: 2000
5. Principal: 0 , Years: 5

```
import java.io.*;
import java.util.*;
class sii
{
public static void main(String args[])
{
try
{
double p,i,si;
int n;
System.out.println("enter the principal amount:");
Scanner t = new Scanner(System.in);
p = t.nextDouble();
System.out.println("enter no.of years:");
n = t.nextInt();
System.out.println("Is senior citizen(y/n):");
char g = t.next().charAt(0);
if (g=='y'||g=='Y')
{
i = 12;
si = (p*n*i)/100;
System.out.println("simple interest is:"+si);
}
if (g=='n'|| g=='N')
{
i=10;
si =(p*n*i)/100;
System.out.println("simple interest is:"+si);
}
if (p<=0)
{
System.out.println("enter the valid amount");
}
if (n<=0)
{
```

```
System.out.println("enter valid no.of years");
}
}
catch (Exception e)
{
System.out.println("invalid");
}
}
}
```

33. Write a program to print the Fibonacci series.

Sample Input:

```
        Enter the n value: 6
```

Sample Output:

```
        0    1        1        2        3        5
```

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
int i=1;
int firstterm = 0;
int secondterm = 1;
int n=10;
while (i<=n)
{
System.out.println(firstterm);
int nextterm = firstterm + secondterm;
firstterm = secondterm;
secondterm = nextterm;
i++;
}
}
catch (Exception e)
{
System.out.println("invalid");
}
}
}
```

34. Java Program to Find Even Sum of Fibonacci Series Till number N?

Sample Input: n = 4
Sample Output: 33

(N = 4, So here the fibonacci series will be produced from 0th term till 8th term:0, 1, 1, 2, 3, 5, 8, 13, 21
Sum of numbers at even indexes = 0 + 1 + 3 + 8 + 21 = 33)

```
import java.util.Scanner;
import java.io.*;
public class rec {
  public static void main(String[] args){
    int my_input, i, sum;
        Scanner my_scanner = new Scanner(System.in);
        System.out.println("Enter the value of N: ");
    my_input = my_scanner.nextInt();
    int fabonacci[] = new int[2 * my_input + 1];
    fabonacci[0] = 0;
    fabonacci[1] = 1;
    sum = 0;
    for (i = 2; i <= 2 * my_input; i++) {
      fabonacci[i] = fabonacci[i - 1] + fabonacci[i - 2];
      if (i % 2 == 0)
        sum += fabonacci[i];
    }
    System.out.printf("Even sum of fibonacci series till number %d is %d" , my_input, sum);
  }
}
```

35. Write a program to print the numbers from M to N by skipping K numbers in between?

Sample Input:

M = 50
N = 100
K = 7

Sample Output:

50, 58, 66, 74, …..

Test cases:

1. M = 15, N = 05, K = 02
2. .M = 25, N = 50, K = 04
3. M = 15, N = 100, K = -02
4. M = 0 , N = 0 , K = 2
5. M = 200 , N = 200 , K = 50

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
int m,n,k;
System.out.println("M=");
Scanner p = new Scanner(System.in);
m = p.nextInt();
System.out.println("N=");
n = p.nextInt();
System.out.println("K=");
k = p.nextInt();
if (m<=0 || n<=0 || k<=0)
System.out.println("invalid input");
}
else if (m<=n || k>=n || n<=m)
{
System.out.println("invalid input");
}
while(m<=n)
{
System.out.println(m);
m = m+k+1;
}
}
catch (Exception e)
{
System.out.println("invalid");
}
}
}
```

36. Write a program to print all the composite numbers between a and b?

Sample Input:

A = 12

B = 19

Sample Output

14, 15, 16, 18

Test cases:

1. A = 11, B = 11
2. A = 20, B = 10
3. A = 0, B = 0
4. A = -5, B = 5
5. A = 7, B = -12

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

```
class rec
{
public static void main(String args[])
{
int a,b;
Scanner s = new Scanner(System.in);
System.out.println("enter a value:");
a = s.nextInt();
System.out.println("enter b value:");
b = s.nextInt();
System.out.println("composite numbers between" +a+ "and" +b+ "are:");
printComposite(a,b);
}
private static boolean  findComposite(int num)
{
if (num <=1)
{
return false;
}
for ( int i=2;i<=Math.sqrt(num);i++)
{
if (num % i == 0)
{
return true;
}
}
return false;
}
public static void printComposite(int a,int b)
{
for(int i =a;i<=b;i++)
{
if(findComposite(i))
{
System.out.println(i + " ");
}
}
System.out.println();
}
}
```

37. Find the factorial of n?

Sample Input:

N = 4

Sample Output:

4 Factorial = 24

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = Q
5. N = 3A

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
int fact = 1;
int n=5;
int i=1;
while(i<=n)
{
fact = fact * i;
i++;
}
System.out.printf("factorial of %d = %d",n ,fact);
}
catch (Exception e)
{
System.out.println("invalid");
}
}
}
```

38. Find the year of the given date is leap year or not

Sample Input:

Enter Date: 04/11/1947

Sample Output:

Given year is Non Leap Year

Test cases:

1. 04/11/19.47
2. 11/15/1936
3. 31/45/1996
4. 64/09/1947
5. 00/00/2000

39. Find the number of factors for the given number

Sample Input:

Given number: 100

Sample Output:

Number of factors = 9

Test cases:

1. 343
2. 1080
3. -243
4. 101010
5. 0

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
int count = 0;
int n;
System.out.println("enter n value:");
Scanner p = new Scanner(System.in);
n = p.nextInt();
for (int i=1; i<=n ;i++)
{
if (n % i == 0)
{
count++;
}
}
System.out.println("no.of factors are:"+count);
}
catch (Exception e)
{
System.out.println("invalid");
}
}
}
```

40. Write a program to print the given number is Perfect number or not?

Sample Input:

Given Number: 6

Sample Output:

It’s a Perfect Number

Test cases:

1. 17

2. 26!
3. 143
4. 84.1
5. -963

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
int n,sum=0;
System.out.println("enter n value:");
Scanner s = new Scanner(System.in);
n = s.nextInt();
for (int i=1;i<n;i++)
{
if (n%i == 0)
{
sum = sum + i;
}
}
if (sum == n)
{
System.out.println("perfect number");
}
else
{
System.out.println("not a perfect number");
}
}
catch (Exception e)
{
System.out.println("invalid");
}
}
}
```

41. Write a program to find the square, cube of the given decimal number

Sample Input:

Given Number: 0.6

Sample Output:

Square Number: 0.36
Cube Number:0.216

Test cases:

1. 12
2. 0
3. -0.5
4. 14.25
5. -296

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
System.out.println("enter the decimal number:");
Scanner s = new Scanner(System.in);
float n = s.nextFloat();
System.out.println("square of the number is:"+n*n);
System.out.println("cube of:" +n+ "is:" +(n*n*n));
}
catch (Exception e)
{
System.out.println("invalid");
}
}
}
```

42. Find the nth odd number after n odd number

Sample Input: N : 7

Sample Output:

Hence the values printed for i are 1 , 3 , 5.

Test cases:

1. N = 0
2. N = -6
3. N = 2021
4. N = -14.5
5. N = -196

40 Program to find the frequency of each element in the array.

Sample Input & Output:

{1, 2, 8, 3, 2, 2, 2, 5, 1}

Pseudo:

```
Element | Frequency
--------------------------
1          |        2
2          |        4
8          |        1
3          |        1
4         |        1
```

43. Program to find whether the given number is Armstrong number or not

Sample Input:

Enter number: 153

Sample Output:

Given number is Armstrong number

Test cases:

1. 370
2. 1
3. 371
4. 145678
5. 0.21345

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
int n;
System.out.println("enter the number:");
Scanner p = new Scanner(System.in);
n = p.nextInt();
int temp = n;
int r,sum=0;
while (n>0)
{
r = n % 10;
n = n/10;
sum = sum+r*r*r;
}
if (temp == sum)
{
System.out.println("it is an armstrong number");
}
else
{
System.out.println("it is not an armstrong number");
}
}
catch (Exception e)
{
System.out.println("invalid input");
}
}
}
```

44. Write a program to find the sum of digits of N digit number (sum should be single digit)

Sample Input:

Enter N value: 3

Enter 3 digit numbers: 143

Test cases:

1. N = 2, 158
2. N = 3, 14
3. N = 4, 0148
4. N = 1, 0004
5. N = 4, 7263

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
int n,digit,sum=0;
System.out.println("enter the value of n:");
Scanner p = new Scanner(System.in);
n = p.nextInt();
while (n>0)
{
digit = n % 10;
sum = sum + digit;
n = n/10;
}
System.out.println("sum of digits:"+sum);
}
catch (Exception e)
{
System.out.println("invalid");
}
}
}
```

45. Write a program to find the square root of a perfect square number(print both the positive and negative values)

Sample Input:

Enter the number: 6561

Sample Output:

Square Root: 81, -81

Test cases:

1. 1225
2. 9801
3. 1827
4. -100
5. 0

```
import java.io.*;
import java.util.*;
class mom
{
public static void main(String args[])
{
try
{
int n;
System.out.println("enter a number:");
Scanner s = new Scanner(System.in);
n = s.nextInt();
double x = Math.sqrt(n);
if (x == (int)x)
{
System.out.println("square root of perfect square:"+x);
System.out.print(" ,- "+x);
}
else
{
System.out.println("please enter perfect square number:");
}
}
catch (Exception e)
{
System.out.println("invalid input");
}
}
}
```

46. Write a program to given an integer n, return true if it is a power of three. Otherwise, return false.

Input =27
Output= true
Explanation: 27=3^3

Test cases:

1. 12
2. abc@45
3. 1827
4. -100
5. 0

47. Write a program to given a string paragraph and a string array of the banned words banned, return the most frequent word that is not banned. It is guaranteed there is at least one word that is not banned, and that the answer is unique.

Input Paragraph=”Ram hit a ball, the hit ball flew far after it was hit”,
Banned = [hit]
Output=”Ball”

48. Write a program to given a fixed-length integer array arr, duplicate each occurrence of zero, shifting the remaining elements to the right.
Input: arr = [1, 0, 2, 3, 0, 4, 5, 0]
Output: [1, 0, 0, 2, 3, 0, 0, 4]
Explanation: After calling your function, the input array is modified to [1, 0, 0, 2, 3, 0, 0, 4]

49. Write a program to given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.
Input nums = [3, 0, 1]
Output: 2
Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0, 3]. 2 is the missing number in the range since it does not appear in nums.

50. Write a program to given an integer array nums, find the subarray with the largest sum, and return its sum.
Input nums = [-2, 1,-3, 4,-1, 2, 1,-5, 4]
Output: 6
Explanation: The subarray [4,-1, 2, 1] has the largest sum 6.

51. Write a program to print the multiplication table of number m up to n.

Sample Input:
M = 4
N = 5

Sample Output:
1x4=4
2x4=8
3x4=12

4x4=16
5x4=20

Test cases:

M = 6, N = -3
M = -3, N = 5
M = 4, N = 0
M = 0, N = 0
M = -5, N = -5

52. Write Java programs to implement multiple threads and apply join method for thread and thread has to be started after 500ms using sleep ().

53. Generate a Java code that implements java selection and iteration statements. Use do while loop to process a menu selection. When a menu is selected, it should display the syntax of the selected statements.

54. Create a simple generics class with type parameters for sorting values of different types.

55. Create a class name ‘overload’. Write a program to assign the values for two values by different number of arguments using a single function.

56. Write a Java Program to count the number of words in a string using Hash Map.

57. Write a Java Program to read an email and password from excel sheet by retrieving the cell using getRow() and getCell() method.

58. Write a Java program to sorts the given value using Hash Map.

59. Write a Java program to find distinct characters and their count in a string.

60. Write a program to print all the unique characters in a String. For instance, if the input string is “abcb”, the output will be the characters ‘a’ and ‘c’ as they are unique. The character ‘b’ repeats twice and so it will not be printed.

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

INSTITUTE OF PLACEMENT AND TRAINING

CSA09 –JAVA PROGRAMMING

1. Write a Java Program to Convert a Given Number of Days in Terms of Years, Weeks & Days.

   Sample Input&Output::

   Enter the number of days: 756

   No. of years: 2
   No. of weeks: 3
   No. of days: 5

   Test cases:

   1. 38
   2. 3.6
   3. 0
   4. -365
   5. -45

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int d,x;
int w,z,y;
Scanner s = new Scanner(System.in);
System.out.println("enter no.of days:");
d = s.nextInt();
y = d/365;
System.out.println("no.of years:"+y);
x = d % 365;
d = x % 7;
w = x / 7;
System.out.println("days:"+d);
System.out.println("weeks:"+w);
}
}
```

Given a date, return the corresponding day of the week for that date.
The input is given as three integers representing the day, month and year respectively.
Return the answer as one of the following values {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}.

Example 1:

Input: day = 31, month = 8, year = 2019
Output: "Saturday"

Example 2:

Input: day = 18, month = 7, year = 1999
Output: "Sunday"

Example 3:

Input: day = 15, month = 8, year = 1993

Output: "Sunday"

Constraints:

- The given dates are valid dates between the years 1971 and 2100.

2. Write a program to find the number of student users in the college, get the total users, staff users details from the client. Note for every 3 staff user there is one Non teaching staff user assigned by default.

   Sample Input:

   Total Users: 856
   Staff Users: 126

   Sample Output:

   Student Users: 688

   Test Cases:

   1. Total User: 0
   2. Total User: -143
   3. Total User: 1026, Staff User: 1026
   4. Total User: 450, Staff User: 540
   5. Total User: 600, Staff User: 450

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int t,s,stu,nt;
Scanner p = new Scanner(System.in);
System.out.println("enter total users:");
t = p.nextInt();
System.out.println("enter staff users:");
s = p.nextInt();
nt = s/3;
stu = t-s-nt;
System.out.println("total no.of student users:"+stu);
}
}
```

3. Write a program to print number of factors and to print nth factor of the given number.

   Sample Input:

   Given Number: 100

   N = 4

   Sample Output:

   Number of factors = 9
   4th factor of 100 = 5

   Test Cases:

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

INSTITUTE OF PLACEMENT AND TRAINING

CSA09 –JAVA PROGRAMMING

1. Given Number = 512 , N = 6
2. Given Number = 343 , N = 7
3. Given Number = 1024 , N = 0
4. Given Number = -6561 , N = 3
5. Given Number = 0 , N = 2

```
import java.util.Scanner;

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

        // Input: Given number and n
        System.out.print("Given Number: ");
        int givenNumber = scanner.nextInt();

        System.out.print("N = ");
        int n = scanner.nextInt();

        // Calculate factors and nth factor
        int factorCount = calculateFactorCount(givenNumber);
        int nthFactor = calculateNthFactor(givenNumber, n);

        // Output the result
        System.out.println("Number of factors = " + factorCount);
        System.out.println(n + "th factor of " + givenNumber + " = " + nthFactor);

        scanner.close();
    }

    // Function to calculate the number of factors
    private static int calculateFactorCount(int number) {
        int count = 0;
        for (int i = 1; i <= number; i++) {
            if (number % i == 0) {
                count++;
            }
        }
        return count;
    }

    // Function to calculate the nth factor
    private static int calculateNthFactor(int number, int n) {
        int count = 0;
        for (int i = 1; i <= number; i++) {
            if (number % i == 0) {
                count++;
                if (count == n) {
                    return i;
                }
            }
        }
        return -1; // Return -1 if the nth factor doesn't exist
    }
}
```

4. Write a program to print n prime numbers after nth Prime number

   Sample Input:

   N = 3

   Sample Output:

   3rd Prime number is 5

   3 prime numbers after 5 are: 7, 11, 13

   Test cases:

   1. N = P
   2. N = 0
   3. N = -4
   4. N = 11
   5. N = 7.2

```
import java.util.Scanner;

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

        // Input
        System.out.print("Enter the value of N: ");
        int N = scanner.nextInt();

        // Calculate Nth prime number
        int nthPrime = getNthPrime(N);
        System.out.println(N + "rd Prime number is " + nthPrime);

        // Print N prime numbers after the Nth prime number
        for (int i = 0; i < N; i++) {
            nthPrime = getNextPrime(nthPrime);
            System.out.print(nthPrime + ", ");
        }
    }

    // Method to check if a number is prime
    private static boolean isPrime(int num) {
        if (num < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }

    // Method to get the next prime number
    private static int getNextPrime(int start) {
        int num = start + 1;
        while (!isPrime(num)) {
            num++;
        }
        return num;
    }

    // Method to get the Nth prime number
    private static int getNthPrime(int N) {
        int count = 0;
        int num = 1;
        while (count < N) {
            num = getNextPrime(num);
            count++;
        }
        return num;
    }
}
```

5. Write a Program to create a list of all numbers in a range which are perfect squares and the sum of the digits of the number is less than 10.

   Sample Input & Output:

   Enter lower range: 1
   Enter upper range: 40
   [1, 4, 9, 16, 25, 36]

Test case:

1. Enter lower range: 50
   Enter upper range: 100
2. Enter lower range: 5
   Enter upper range: 8
3. Enter lower range: 10
   Enter upper range: 5
4. Enter lower range: 500
   Enter upper range: 500
5. Enter lower range: 0

```
                Enter upper range: -100
```

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

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

      System.out.println("Enter the lower range:");
      int l = s.nextInt();

      System.out.println("Enter the upper range:");
      int u = s.nextInt();

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

      for (int i = l; i <= u; i++) {
         if (i > Math.sqrt(u)) {
            // Break the loop if the number exceeds the square root of the upper limit
            break;
         }
         int square = i * i;
         if (digitSumLessThan10(square)) {
            perfectSquares.add(square);
         }
      }

      System.out.println(perfectSquares);
   }

   private static boolean digitSumLessThan10(int num) {
      int sum = 0;
      while (num > 0) {
         sum += num % 10;
         num /= 10;
      }
      return sum < 10;
   }
}
```

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int l,u;
Scanner s = new Scanner(System.in);
System.out.println("enter the l:");
l = s.nextInt();
System.out.println("enter the u:");
u = s.nextInt();
int sum = 0;
List<Integer> ssquareslist = new ArrayList<>();
for(int i=l;i<=u;i++)
{
int square = i * i;
ssquareslist.add(square);
sum += square;
if(sum>10)
{
break;
}
}
System.out.println(ssquareslist);
}
}
```

```
    import java.io.*;
    import java.util.*;
    class rec
    {
    public static void main(String args[])
    {
    int l,u;
    Scanner s = new Scanner(System.in);
    System.out.println("enter the l:");
    l = s.nextInt();
    System.out.println("enter the u:");
    u = s.nextInt();
    int sum = 0;
    List<String> ssquareslist = new ArrayList<>();
    for(int i=l;i<=u;i++)
    {
    int square = i * i;
    ssquareslist.add("(" +i+ ", " +square+ ")");
    }
    System.out.println(ssquareslist);
    }
    }
```

6. Write a program to print unique permutations of a given number

Sample Input:
Given Number: 143
Sample Output:
Permutations are:
134
143
314
341
413
431
Test cases:
1. 0
2. 111
3. 505
4. -143
5. -598

7. Write a Program to create an array with the First Element as the Number and Second Element as the Square of the Number.
Sample Input:
Enter the lower range:45
Enter the upper range:49
Sample Output:
[(45, 2025), (46, 2116), (47, 2209), (48, 2304), (49, 2401)]
Test case:
1. Enter lower range: 50
Enter upper range: 100
2. Enter lower range: 5
Enter upper range: 8
3. Enter lower range: 10
Enter upper range: 5
4. Enter lower range: 500
Enter upper range: 500
5. Enter lower range: 0
Enter upper range: -100

```
import java.util.Scanner;

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

      System.out.println("Enter the lower range:");
      int l = s.nextInt();

      System.out.println("Enter the upper range:");
      int u = s.nextInt();
```

```
        for (int i = l; i <= u; i++) {
            int square = i * i;
            System.out.println("Element: " + i + ", Square: " + square);
}
}
}
```

8. Develop a JAVA code to display the balance. Include the following members:

- Design a class to represent a bank account.
- **Data Members:** Name of the depositor, Account number, Type of account(Savings/Current), Balance amount in the account(Minimum balance is Rs.500.00)
- **Methods:**
    1. To read account number, Depositor name, Type of account.
    2. To deposit an amount (Deposited amount should be added with it)
    3. To withdraw an amount after checking balance(Minimum balance must be Rs.500.00

Note : Assume that balance amount = 10000

Test Cases

1. 100, Raja, S, 8000
2. Raja, 100, S, 9000
3. 101, Rani, S, 12000
4. 102, Ragu, W, 8000
5. 103, Ravi, C, 10000

```
import java.io.*;
import java.util.*;
class Bankaccount
{
int accountNum,balance;
String depositername,Accounttype;
Scanner s = new Scanner(System.in);
void getdata()
{
System.out.println("enter the account number:");
accountNum = s.nextInt();
System.out.println("enter balance:");
balance = s.nextInt();
System.out.println("enter depositername:");
depositername = s.nextLine();
System.out.println("enter account type:");
Accounttype = s.nextLine();
double amount;
System.out.println("enter the amount:");
amount = s.nextDouble();
}
void deposit()
{
if(amount>0)
{
balance += amount;
System.out.println("amount deposited succesfully");
}
else
{
System.out.println("invalid deposite amount");
}
}
void withdraw()
{
if (amount > 0)
{
if (balance - amount >= 500)
{
balance -= amount;
System.out.println("Amount withdrawn successfully. New balance: Rs. " + balance);
}
else
{
System.out.println("Insufficient balance. Minimum balance should be Rs. 500.");
}
}
else
{
System.out.println("Invalid withdrawal amount.");
}
}
class rec
{
public static void main(String args[])
{
Bankaccount a = new Bankaccount();
a.getdata();
a.deposit();
a.withdraw();
}
}
}
```

9. Develop a code to Reverse and Add a Number until you get a Palindrome.

Sample Input If 7325 is input number, then

7325 (Input Number) + 5237 (Reverse Of Input Number) = 12562

```
12562 + 26521 = 39083
39083 + 38093 = 77176
77176 + 67177 = 144353
144353 + 353441 = 497794 (Palindrome)
```

Test Cases

1. 8765
2. -8765
3. 0
4. EIGHT FIVE
5. 87.57

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int num;
Scanner s = new Scanner(System.in);
System.out.println("enter the number:");
num = s.nextInt();
int reverse = reverseNumber(num);
int sum = num + reverse;
while(!ispalindrome((sum)))
{
System.out.println(num+"+"+reverse+"="+sum);
num = sum;
reverse = reverseNumber(num);
sum = num + reverse;
}
System.out.println(num+"+"+reverse+"="+sum+"(Palindrome)");
}
private static int reverseNumber(int number)
{
int reversed = 0,rem;
while(number!=0)
{
rem = number % 10;
reversed = reversed * 10 + rem;
number = number / 10;
}
return reversed;
}
private static boolean ispalindrome(int number)
{
return number == reverseNumber(number);
}
```

```
    }
```

10. Create Customer class with deposit() and withdraw() as synchronized methods. Declare AccountNo, AccName and Balance as Instance Variables inside the class. From the main class, Input the amount for withdraw() operation and if requested amount is not available in existing Balance amount, withdraw() method should be temporarily suspended using wait() method until deposit() method receives the input for amount, to be added in the existing Balance amount and then withdraw() would be completed in a successful manner. Develop the above scenario using Synchronization and Inter-Thread Communication.

Note : existing Bank balance amount 10000
Sample Input : 12000, 3000
Sample Output : Withdraw operation success, balance amount 1000

Test Cases

1. 11000, 4000
2. -10000, -2000
3. 0, 0
4. EIGHT SEVEN, FIVE
5. 100.67, 200.68

11. Given an integer n, return a string array answer (1-indexed) where:

answer[i] == "FizzBuzz" if i is divisible by 3 and 5.
answer[i] == "Fizz" if i is divisible by 3.
answer[i] == "Buzz" if i is divisible by 5.
answer[i] == i (as a string) if none of the above conditions are **true.**

Example 1:
Input: n = 3
Output: ["1","2","Fizz"]

Test Case

Test Case	Inputs
1.	n = 5
2.	n = 10
3.	n = 12
4.	n = 18
5.	n = 20

12. Write a Java program to find the common elements in two array of Positive integer

Sample Input:
[1, 2, 3, 4]
[2, 4, 5, 6, 7]
Expected output: [2, 4]

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

```
public class rec {
    public static void main(String args[]) {
        int x[] = new int[4];
        int y[] = new int[5];
        Scanner s = new Scanner(System.in);

        System.out.println("Enter the 1st array elements:");
        for (int i = 0; i < 4; i++) {
            x[i] = s.nextInt();
        }

        System.out.println("Enter the 2nd array elements:");
        for (int i = 0; i < 5; i++) {
            y[i] = s.nextInt();
        }

        System.out.println("Resulting array:");
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 5; j++) {
                if (x[i] == y[j]) {
                    System.out.print(x[i] + " ");
                }
            }
        }
    }
}
```

Test Case

Test Case	Inputs-1	Inputs-2
1.	[1, 2, 3, 4]	[4,5,6,7,8]
2.	[a, b, c, d]	[a, b, c, d]
3.	[1, -2, 3, 4]	[1,-2,5,7,8]
4.	[@, #, 34, 45]	[@,#,%,$,]
5.	[45,78,56,89]	[92,34,56,-78,-90]

13. Given a string s consisting of words and spaces, return *the length of the* **last** *word in the string.* A **word** is a maximal substring consisting of non-space characters only. There will be at least one word, consists of only English letters and spaces ' '.
    Example 1:
    Input: s = "Hello World"
    Output: 5
    Explanation: The last word is "World" with length 5.

    Test Case

Test Case	Inputs-1
1.	Maximal Substring Consisting
2.	**lea@st one wor2d**
3.	1254 98076
4.	& * ( ) % # $
5.	letters and spaces

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
String input;
Scanner s = new Scanner(System.in);
System.out.println("enter the input:");
input = s.nextLine();
String word = input.trim();
int lengthoflastword = getLengthOfLastWord(word);
System.out.println("Length of the last word: " +lengthoflastword);
}
private static int getLengthOfLastWord(String input) {
        int lastSpaceIndex = input.lastIndexOf(' ');
        String lastWord = input.substring(lastSpaceIndex + 1);
        return lastWord.length();
    }
}
```

14. Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

- I can be placed before V (5) and X (10) to make 4 and 9.
- X can be placed before L (50) and C (100) to make 40 and 90.

- C can be placed before D (500) and M (1000) to make 400 and 900. Given a roman numeral, convert it to an integer.

Example:
Input: s = "III"
Output: 3

Test Case	Inputs
1.	LVIII
2.	MCMXCI
3.	V
4.	LZAII
5.	MCCDTIV

15. Given two strings ransomNote and magazine, return true if ransomNote can be constructed by using the letters from magazine and false otherwise. Each letter in magazine can only be used once in ransomNote.

Example 1:
Input: ransomNote = "a", magazine = "b"
Output: false
Test Case

Test Case	Inputs
1.	ransomNote = "aa", magazine = "ab"
2.	ransomNote = "aa", magazine = "aab"
3.	ransomNote = "abc", magazine = "abc"
4.	ransomNote = "good", magazine = "better"
5.	ransomNote = "xyz", magazine = "123"

16. You are given an m x n binary matrix mat of 1's (representing soldiers) and 0's (representing civilians). The soldiers are positioned in front of the civilians. That is, all the 1's will appear to the left of all the 0's in each row.

A row i is weaker than a row j if one of the following is true:

The number of soldiers in row i is less than the number of soldiers in row j.

Both rows have the same number of soldiers and i< j. Return the indices of the k weakest rows in the matrix ordered from weakest to strongest.

Example 1:
Input: mat =
[[1,1,0,0,0],
 [1,1,1,1,0],
 [1,0,0,0,0],
 [1,1,0,0,0],
 [1,1,1,1,1]],
k = 3
Output: [2,0,3]
Explanation:

The number of soldiers in each row is:
- Row 0: 2
- Row 1: 4
- Row 2: 1
- Row 3: 2
- Row 4: 5

The rows ordered from weakest to strongest are [2,0,3,1,4].

Example 2:

Input: mat =
[[1, 0, 0, 0],
[1, 1, 1,1],
[1, 0, 0, 0],
[1, 0, 0,0]],
k = 2

Output: [0,2]

Explanation:

The number of soldiers in each row is:
- Row 0: 1
- Row 1: 4
- Row 2: 1
- Row 3: 1

The rows ordered from weakest to strongest are [0, 2, 3, 1].

17. Given an integer num, return the number of steps to reduce it to zero. In one step, if the current number is even, you have to divide it by 2, otherwise, you have to subtract 1 from it.

Example 1:

Input: num = 14

Output: 6

Explanation:

Step 1) 14 is even; divide by 2 and obtain 7.
Step 2) 7 is odd; subtract 1 and obtain 6.
Step 3) 6 is even; divide by 2 and obtain 3.
Step 4) 3 is odd; subtract 1 and obtain 2.
Step 5) 2 is even; divide by 2 and obtain 1.
Step 6) 1 is odd; subtract 1 and obtain 0.

Test Case

Test Case	**Inputs**
1.	n = 5
2.	n = 10
3.	n = 12
4.	n = 18

5.	n = 20

```
public class ReduceToZero {
    public static void main(String[] args) {
        int num = 14;

        int steps = numberOfSteps(num);

        System.out.println("Number of steps to reduce to zero: " + steps);  // Output: 6
    }

    public static int numberOfSteps(int num) {
        int steps = 0;

        while (num > 0) {
            if (num % 2 == 0) {
                num /= 2;  // If the current number is even, divide it by 2
            } else {
                num -= 1;  // If the current number is odd, subtract 1
            }

            steps++;
        }

        return steps;
    }
}
```

18. Develop a programme that uses Multiple Inheritance concepts to compute a student's grades in six subjects. The total and aggregate are then calculated, and the student's grade is displayed. If the student achieves an aggregate of more than 75%, the grade is Distinction. If the aggregate is between 60 and 75, the grade is First Division. If the aggregate is between 50 and 60, the grade is Second Division. If the aggregate is between 40 and 50, the grade is Third Division. Otherwise, the grade is FAIL.
Sample Input & Output:

```
Enter the marks in python: 90
Enter the marks in c programming: 91
Enter the marks in Mathematics: 92
Enter the marks in Physics: 93
Enter the marks in Chemistry: 92
Enter the marks in Professional Ethics: 93
Total= 551
Aggregate = 91.83
Class: DISTINCTION
```

Test Case

Test Case	**Inputs**
1.	18, 76,93,65,63,98
2.	73,78,79,75,87,0
3.	98,106,120,95,98,34
4.	96,73, -85,95,84,98
5.	78,59.8,76,79,97,67

19. Write a program to calculate tax given the following conditions:
    i. If income is less than or equal to 2,50,000 then no tax
    ii. If taxable income is 2,50,001 – 5,00,000 the charge 10% tax
    iii. If taxable income is 5,00,001 – 10,00,000 the charge 20% tax
    iv. If taxable income is above 10,00,001 then charge 30% tax

Sample Input:

Enter the income: 600000

Sample Output:

Taxable Income: 350000

Tax= 35000

Test Case

Test Case	**Inputs**
1.	400700
2.	2789239
3.	150000
4.	00000
5.	-125486

20. Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading. Compute the bill amount using the following tariff.

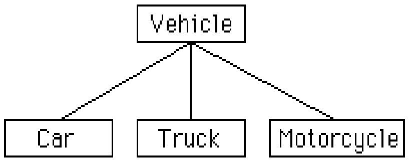
First 100 units – Rs. 1 per unit

101-200 units – Rs. 2.50 per unit

201 -500 units – Rs. 4 per unit

> 501 units – Rs. 6 per unit

21. Design class called Vehicle, which has two subclasses called Car and Truck. The Vehicle class is the superclass of Car, Truck and Motorcycle. The Vehicle class can contain those fields and methods that all Vehicles need (e.g. a license plate, owner etc.), whereas Car, Truck and Motorcycle can contain the fields and methods that are specific to Car, Truck and Motorcycle. Write a Java program to implement the concept of inheritance based on the following diagram



22. Write a program to create a directory that contains the following information.

(a) Name of a person

(b) Address

(c) Telephone Number (if available with STD code)

(d) Mobile Number (if available)
(e) Head of the family
(f) Unique ID No.

23. Write a Java program to create multiple threads for different calculator operations
24. Given a string S partition S such that every substring of the partition is a palindrome. Return the minimum cuts needed for a palindrome partitioning of S
    Input S=”aabbc”
    Output=2
    Explanation: The palindrome partitioning ["aa", "bb", "c"]] could be produced using 2 cut.

25. Design a java code an array arr which consists of only zeros and ones, divide the array into three non-empty parts such that all of these parts represent the same binary value.
    If it is possible, return any [i, j] with i + 1 < j, such that:
    arr[0], arr[1], ..., arr[i] is the first part,
    arr[i + 1], arr[i + 2], ..., arr[j - 1] is the second part, and
    arr[j], arr[j + 1], ..., arr[arr.length - 1] is the third part.
    All three parts have equal binary values.
    If it is not possible, return [-1, -1].

    Note that the entire part is used when considering what binary value it represents. For example, [1, 1, 0] represents 6 in decimal, not 3. Also, leading zeros are allowed, so [0,1,1] and [1,1] represent the same value.

26. Design a special dictionary that searches the words in it by a prefix and a suffix. Implement the WordFilter class: WordFilter (string [] words) Initializes the object with the words in the dictionary. f(string pref, string suff) Returns the index of the word in the dictionary, which has the prefix pref and the suffix suff. If there is more than one valid index, return the largest of them. If there is no such word in the dictionary, return -1.

27. Given an integer array nums, handle multiple queries of the following types: Update the value of an element in nums. Calculate the sum of the elements of nums between indices left and right inclusive where left <= right.Implement the NumArray class: NumArray(int[] nums) Initializes the object with the integer array nums.void update(int index, int val) Updates the value of nums[index] to be val.int sumRange(int left, int right) Returns the sum of the elements of nums between indices left and right inclusive (i.e. nums[left] + nums[left + 1] + ... + nums[right]).
    Input
    ["NumArray", "sumRange", "update", "sumRange"]
    [[[1, 3, 5]], [0, 2], [1, 2], [0, 2]]
    Output
    [null, 9, null, 8]
    Explanation
    NumArray numArray = new NumArray([1, 3, 5]);
    numArray.sumRange(0, 2); // return 1 + 3 + 5 = 9
    numArray.update(1, 2);   // nums = [1, 2, 5]
    numArray.sumRange(0, 2); // return 1 + 2 + 5 = 8

28. Given an integer array arr, return the number of distinct bitwise ORs of all the non-empty subarrays of arr. The bitwise OR of a subarray is the bitwise OR of each integer in the subarray. The bitwise OR of a subarray of one integer is that integer. A subarray is a contiguous non-empty sequence of elements within an array.

Input: arr = [1, 1, 2]
Output: 3
Explanation: The possible subarrays are [1], [1], [2], [1, 1], [1, 2], [1, 1, 2].
These yield the results 1, 1, 2, 1, 3, 3.
There are 3 unique values, so the answer is 3.

29. Given two numbers arr1 and arr2 in base -2, return the result of adding them together. Each number is given in array format: as an array of 0s and 1s, from most significant bit to least significant bit. For example, arr = [1, 1, 0, 1] represents the number (-2)^3 + (-2)^2 + (-2)^0 = -3. A number arr in array, format is also guaranteed to have no leading zeros: either arr == [0] or arr[0] == 1. Return the result of adding arr1 and arr2 in the same format: as an array of 0s and 1s with no leading zeros.
Input: arr1 = [1,1,1,1,1], arr2 = [1,0,1]
Output: [1,0,0,0,0]
Explanation: arr1 represents 11, arr2 represents 5, the output represents 16.

30. You are given a string of digits num, such as "123456579". We can split it into a Fibonacci-like sequence [123, 456, 579]. Formally, a Fibonacci-like sequence is a list f of non-negative integers such that: 0 <= f[i] < 231, (that is, each integer fits in a 32-bit signed integer type), f.length >= 3, and f[i] + f[i + 1] == f[i + 2] for all 0 <= i < f.length - 2. Note that when splitting the string into pieces, each piece must not have extra leading zeroes, except if the piece is the number 0 itself. Return any Fibonacci-like sequence split from num, or return [] if it cannot be done.

Example-1
Input: num = "1101111"
Output: [11, 0, 11, 11]
Explanation: The output [110, 1, 111] would also be accepted.

Example-2
Input: num = "112358130"
Output: []
Explanation: The task is impossible.

Example-3
Input: num = "0123"
Output: []
Explanation: Leading zeroes are not allowed, so "01", "2", "3" is not valid.

31. Create a class date with day, month and year as members. Write appropriate member functions. Create other class students, which have id, name, date of birth and marks of 3 subjects as members. Write appropriate constructor for the student which assigns values to the members. Display the student details in a proper format.
32. Develop a code to Student information to perform the operation like insert, retrieve and remove the record using vector(Student Name, Roll number, Department , Course, Contact information..)
33. Define an Employee class with suitable attributes having get Salary() method, which returns salary withdrawn by a particular employee. Write a class Manager which extends a class Employee, override the gets alary() method, which will return salary of manager by adding traveling _allowance, house rent allowance etc.
34. Create an abstract class Shape. Derive three classes sphere, cone and cylinder from it. Calculate area and volume of all. (Use Method overriding)
35. Write a program to show the employee details using setter and getter methods.
36. Write a JAVA program which will generate the threads:
    - To display 10 terms of Fibonacci series.
    - To display 1 to 10 in reverse order.
37. Write a program to add, retrieve and remove the element from the Array List for employee details(Employee name, Employee ID, Employee Salary, EmployeePF, Employee Allowence),

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

INSTITUTE OF PLACEMENT AND TRAINING

CSA09 –JAVA PROGRAMMING

38. Write program to search key and value from HashTable. (HashTable shows how to get the all keys as Enumeration object. The put () method is used to add the elements in the HashTable. By using Enumeration methods like hasMoreElements () and nextElement () we can read all values from Hashtable. The contains Key ( ) is used for checking the availability of the elements.)
39. Write a Package marks which has one class Student. Accept student detail through parameterized constructor. Write display () method to display details. Create a main class which will use package and calculate total marks and percentage.
40. Write a complex program to illustrate how the thread priorities? Imagine that the first thread has just begun to run, even before it has a chance to do anything. Now comes the higher priority thread that wants to run as well. Now the higher priority thread has to do its work before the first thread starts.

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

INSTITUTE OF PLACEMENT AND TRAINING

CSA09 –JAVA PROGRAMMING

1. Write a program to count all the prime and composite numbers entered by the user.

Sample Input:

Enter the numbers

4
54
29
71
7
59
98
23

Sample Output:

Composite number:3
Prime number:5

Test cases:

1. 33, 41, 52, 61,73,90
2. TEN, FIFTY, SIXTY-ONE, SEVENTY-SEVEN, NINE
3. 45, 87, 09, 5.0 ,2.3, 0.4
4. -54, -76, -97, -23, -33, -98
5. 45, 73, 00, 50, 67, 44

```
import java.util.Scanner;

class PrimeCompositeCounter {
    public static void main(String args[]) {
        int n = 8;
        int primeCount = 0;
        int compositeCount = 0;
        int x[] = new int[8];

        Scanner s = new Scanner(System.in);
        System.out.println("Enter the number:");
        n = s.nextInt();

        System.out.println("Enter " + n + " numbers:");
        for (int i = 0; i < n; i++) {
            x[i] = s.nextInt();
            if (isPrime(x[i])) {
                primeCount++;
            } else {
                compositeCount++;
            }
        }

        System.out.println("Number of prime numbers: " + primeCount);
        System.out.println("Number of composite numbers: " + compositeCount);
    }

    private static boolean isPrime(int num) {
        if (num < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

2. Find the Mth maximum number and Nth minimum number in an array and then find the sum of it and difference of it.

Sample Input:
Array of elements = {14, 16, 87, 36, 25, 89, 34}
M = 1
N = 3

Sample Output:
1stMaximum Number = 89
3rdMinimum Number = 25
Sum = 114
Difference = 64

Test cases:

1. {16, 16, 16 16, 16}, M = 0, N = 1
2. {0, 0, 0, 0}, M = 1, N = 2
3. {-12, -78, -35, -42, -85}, M = 3 , N = 3
4. {15, 19, 34, 56, 12}, M = 6 , N = 3
5. {85, 45, 65, 75, 95}, M = 5 , N = 7

3. Write a program to print the total amount available in the ATM machine with the conditions applied.

Total denominations are 2000, 500, 200, 100, get the denomination priority from the user and the total number of notes from the user to display the total available balance to the user

Sample Input:
Enter the 1st Denomination: 500
Enter the 1st Denomination number of notes: 4
Enter the 2nd Denomination: 100
Enter the 2nd Denomination number of notes: 20
Enter the 3rd Denomination: 200
Enter the 3rd Denomination number of notes: 32

Enter the 4th Denomination: 2000
Enter the 4th Denomination number of notes: 1
Sample Output:
Total Available Balance in ATM: 12400

Test Cases:
3 Hidden Test cases (Think Accordingly based on Denominations)

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int a,a1,b,b1,c,c1,d,d1,x;
Scanner s = new Scanner(System.in);
System.out.println("enter the first denomination:");
a = s.nextInt();
System.out.println("enter the 1st denomination number of notes:");
a1 = s.nextInt();
System.out.println("enter the second denomination:");
b = s.nextInt();
System.out.println("enter the 2nd denomination number of notes:");
b1 = s.nextInt();
System.out.println("enter the third denomination:");
c = s.nextInt();
System.out.println("enter the 3rd denomination number of notes:");
c1 = s.nextInt();
System.out.println("enter the fourth denomination:");
d = s.nextInt();
System.out.println("enter the 4th denomination number of notes:");
d1 = s.nextInt();
x = ((a*a1)+(b*b1)+(c*c1)+(d*d1));
System.out.println("total available balance in ATM:"+x);
}
}
```

4. Write a program using choice to check
    Case 1: Given string is palindrome or not
    Case 2: Given number is palindrome or not
    Sample Input:
    Case = 1
    String = MADAM
    Sample Output:
    Palindrome

Test cases:
1. MONEY
2. 5678765
3. MALAY12321ALAM
4. MALAYALAM
5. 1234.4321

```
import java.io.*;
import java.util.*;
class palin
{
public static void main(String args[])
{
String input="radar";
String reversed_str = "";
for (int i=input.length()-1;i>=0;i--)
{
reversed_str += input.charAt(i);
}
if (input.toLowerCase().equals(reversed_str.toLowerCase()))
{
System.out.println("is a palindrome");
}
else
{
System.out.println("is not a palindrome");
}
}
}

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

    int num = 3553, reversedNum = 0, remainder;

    // store the number to originalNum
    int originalNum = num;

    // get the reverse of originalNum
    // store it in variable
    while (num != 0) {
      remainder = num % 10;
      reversedNum = reversedNum * 10 + remainder;
      num /= 10;
    }
    // check if reversedNum and originalNum are equal
    if (originalNum == reversedNum) {
      System.out.println(originalNum + " is Palindrome.");
    }
    else {
      System.out.println(originalNum + " is not Palindrome.");
    }
  }
}
```

5. Write a program to convert Decimal number equivalent to Binary number and octal numbers?
   Sample Input:
   Decimal Number: 15
   Sample Output:
   Binary Number = 1111
   Octal = 17
      Test cases:
      1. 111
      2. 15.2
      3. 0
      4. B12
      5. 1A.2

```
import java.util.Scanner;
class conversion { public static void main(String[] args)
 {
Scanner in = new Scanner(System.in);
int decimal = in.nextInt();
String binary = Integer.toBinaryString(decimal);
System.out.println(″BINARY IS ″ + binary);
System.out.print(″OCTAL IS ″);
System.out.println(Integer.toOctalString(decimal)); } }
```

6. In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than $10,000 then the employee gets an extra 2% bonus on salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.
   Sample Input & Output:
   Enter the grade of the employee: B
   Enter the employee salary: 50000
   Salary=50000
   Bonus=5000.0

Total to be paid:55000.0

Test cases:

1. Enter the grade of the employee: A
   Enter the employee salary: 8000
2. Enter the grade of the employee: C
   Enter the employee salary: 60000
3. Enter the grade of the employee: B
   Enter the employee salary: 0
4. Enter the grade of the employee: 38000
   Enter the employee salary: A
5. Enter the grade of the employee: B
   Enter the employee salary: -8000

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
char grade;
double salary,bonus=0;
Scanner s = new Scanner(System.in);
System.out.println("enter grade:");
grade = s.next().charAt(0);
System.out.println("enter salary:");
salary = s.nextDouble();
if (grade=='A')
{
bonus = 0.05 * salary;
}
else if(grade == 'B')
{
bonus = 0.10 * salary;
}
if(salary<10000)
{
bonus += 0.02*salary;
}
System.out.println("salary:"+salary);
System.out.println("bonus:"+bonus);
System.out.println("total:"+(salary+bonus));
}
}
```

7. Write a program to print the first n perfect numbers. (Hint Perfect number means **a positive integer that is equal to the sum of its proper divisors)**

Sample Input:
N = 3
Sample Output:
First 3 perfect numbers are: 6 , 28 , 496
Test Cases:
1. N = 0
2. N = 5
3. N = -2
4. N = -5
5. N = 0.2

8. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:
Enter the marks in python: 90
Enter the marks in c programming: 91
Enter the marks in Mathematics: 92
Enter the marks in Physics: 93
Total= 366
Aggregate = 91.5
DISTINCTION

Test cases:

a) 18, 76,93,65
b) 73,78,79,75
c) 98,106,120,95
d) 96,73, -85,95
e) 78,59.8,76,79

```
import java.util.Scanner;

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

        // Get input from the user
        System.out.print("Enter the marks in Python: ");
        int pythonMarks = scanner.nextInt();

        System.out.print("Enter the marks in C Programming: ");
        int cProgrammingMarks = scanner.nextInt();

        System.out.print("Enter the marks in Mathematics: ");
        int mathMarks = scanner.nextInt();

        System.out.print("Enter the marks in Physics: ");
        int physicsMarks = scanner.nextInt();

        // Calculate total and aggregate
        int totalMarks = pythonMarks + cProgrammingMarks + mathMarks + physicsMarks;
        double aggregate = totalMarks / 4.0;

        // Display total and aggregate
        System.out.println("Total = " + totalMarks);
        System.out.println("Aggregate = " + aggregate);

        // Determine and display the grade based on aggregate
        if (aggregate > 75) {
            System.out.println("Grade: Distinction");
        } else if (aggregate >= 60 && aggregate < 75) {
            System.out.println("Grade: First Division");
        } else if (aggregate >= 50 && aggregate < 60) {
            System.out.println("Grade: Second Division");
        } else if (aggregate >= 40 && aggregate < 50) {
            System.out.println("Grade: Third Division");
        } else {
            System.out.println("Grade: Fail");
        }
    }
}
```

9. Write a program to read the numbers until -1 is encountered. Find the average of positive numbers and negative numbers entered by user.

Sample Input:

```
Enter -1 to exit…
Enter the number: 7
Enter the number: -2
Enter the number: 9
Enter the number: -8
Enter the number: -6
Enter the number: -4
Enter the number: 10
Enter the number:  -1
```

Sample Output:

```
The average of negative numbers is: -5.0
The average of positive numbers is : 8.66666667
```

Test cases:

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

INSTITUTE OF PLACEMENT AND TRAINING

CSA09 –JAVA PROGRAMMING

1. -1,43, -87, -29, 1, -9
2. 73, 7-6,2,10,28,-1
3. -5, -9, -46,2,5,0
4. 9, 11, -5, 6, 0,-1
5. -1,-1,-1,-1,-1

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
int p=0,ne=0;
int count=0,count1=0;
float t,r;
int x[]=new int[8];
System.out.println("enter the values:");
Scanner s = new Scanner(System.in);
for(int i=0;i<8;i++)
{
x[i] = s.nextInt();
if (x[i] == -1)
{
break;
}
}
for(int i=0;i<8;i++)
{
if(x[i] >= 0)
{
System.out.println("positive number:"+(p=p+x[i]));
count++;
System.out.println("no.of positive numbers are:"+count);
}
else
{
System.out.println("negative number:"+(ne = ne+x[i]));
count1++;
System.out.println("no.of negative numbers are:"+count1);
}
}
t = p/count;
System.out.println("average of positive numbers:"+t);
r = ne/count1;
System.out.println("average of negative numbers:"+r);
}
}
```

10. Write a program to read a character until a ***** is encountered. Also count the number of uppercase, lowercase, and numbers entered by the users.

    Sample Input:
    Enter * to exit…
    Enter any character: W
    Enter any character: d
    Enter any character: A
    Enter any character: G
    Enter any character: g
    Enter any character: H
    Enter any character: *
    Sample Output:
    Total count of lower case:2
    Total count of upper case:4
    Total count of numbers =0
    Test cases:

    1. 1,7,6,9,5
    2. S, Q, l, K,7, j, M
    3. M, j, L, &, @, G
    4. D, K, I, 6, L, *
    5. *, K, A, e, 1, 8, %, *

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
Scanner a = new Scanner(System.in);
String c ="   ";
int c1=0,c2=0,c3=0;
while(c.charAt(0)!='*')
{
System.out.println("enter any character:");
c = a.next();
if (Character.isUpperCase(c.charAt(0)))
{
c1++;
}
else if (Character.isLowerCase(c.charAt(0)))
{
c2++;
}
else if (Character.isDigit(c.charAt(0)))
{
c3++;
}
}
System.out.println("total no.of upper case:"+c1);
System.out.println("total no.of lower case:" +c2);
System.out.println("total no.of digits:"+c3);
}
}
```

11. Write a program to calculate the factorial of number using recursive function.

Sample Input & Output:

Enter the value of n: 6

Sample Input & Output:

The factorial of 6 is: 720

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = M
5. N = %

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
try
{
int n;
System.out.println("enter the number:");
Scanner s = new Scanner(System.in);
n = s.nextInt();
System.out.println("the factorial of" +n+ "is");
System.out.println(recur(n));
}
catch (Exception e)
{
System.out.println("invalid");
}
}
public static int recur(int x)
{
if (x>0)
{
```

```
return x*recur(x-1);
}
else
{
return 1;
}
}
}
```

12. Write a Program to Find the Nth Largest Number in a array.

Sample Input:

List : {14, 67, 48, 23, 5, 62}

N = 4

Sample Output:

4th Largest number: 23

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = M
5. N = %

13. Write a program to convert the Binary to Decimal, Octal

Sample Input:

Given Number: 1101

Sample Output:

Decimal Number: 13

Octal:15

Test cases:

1. 211
2. 11011
3. 22122
4. 111011.011
5. 1010.0101

```
import java.util.Scanner;

public class BinaryConverter {

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

        // Input binary number from the user
        System.out.print("Enter a binary number: ");
        String binaryInput = scanner.nextLine();
        // Convert binary to decimal and octal
        int decimalValue = Integer.parseInt(binaryInput, 2);
        String octalValue = Integer.toOctalString(decimalValue);

        // Print the results
        System.out.println("Decimal equivalent: " + decimalValue);
        System.out.println("Octal equivalent: " + octalValue);

        scanner.close();
    }
}
```

14. Write a program to find the number of special characters in the given statement

Sample Input:

Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.

Sample Output:

Number of special Characters: 5

```
import java.io.*;
import java.util.*;
class rec
{
public static void main(String args[])
{
String input;
System.out.println("enter the statement:");
Scanner s = new Scanner(System.in);
input = s.nextLine();
int sp_couunt = 0;
for (int i=0;i<input.length();i++)
{
char ch = input.charAt(i);
if (!Character.isLetterOrDigit(ch) && (!Character.isWhitespace(ch)))
{
sp_couunt++;
}
}
System.out.println("number of special characters:"+sp_couunt);
}
}
```

15. Write a Program to Remove the Duplicate Items from a array.

Sample Input:

Enter the number of elements in array:7
Enter element1:10

```
        Enter element2:20
        Enter element3:20
        Enter element4:30
        Enter element5:40
        Enter element6:40
        Enter element7:50
    Sample Output:
        Non-duplicate items:
        [10, 20, 30, 40, 50]
```

16. Bank is a class that provides method to get the rate of interest. But, rate of interest may differ according to banks. For example, SBI, ICICI and AXIS banks are providing 8.4%, 7.3% and 9.7% rate of interest. Write a Java program for above scenario.

    Sample Input SBI, 8.4

    Sample Output

    Test case

    1. SBI, 8.3
    2. ICICI, 7.3
    3. AXIS, 9.7
    4. SBI, 8.6
    5. AXIX, 7.6

```
import java.io.*;
import java.util.*;
class Bank
{
void display()
{
System.out.println("Rate of interests of bank");
}
}
class SBI extends Bank
{
double r=8.4;
void display1()
{
System.out.println("rate of interest of SBI:"+r);
}
}
class ICICI extends Bank
{
double r1=7.3;
void display2()
{
System.out.println("rate of interest of ICICI:"+r1);
}
}
class AXIS extends Bank
{
double r2=9.7;
void display3()
{
System.out.println("rate of interest of AXIS:"+r2);
}
}
class rec
{
public static void main(String args[])
{
AXIS  a = new AXIS();
a.display3();
a.display();
SBI b = new SBI();
b.display1();
ICICI c = new ICICI();
c.display2();
}
}
```

17. Bring out the situation in which member names of a subclass hide members by the same name in the super class. How it can be resolved? Write Suitable code in Java and Implement above scenario with the Parametrized Constructor (accept int type parameter) of the Super Class can be called from Sub Class Using super () and display the input values provided.

Sample Input : 100, 200
Sample Output : 100, 200

Test Cases

1. 10, 20
2. -20, -30
3. 0, 0
4. EIGHT FIVE
5. 10.57, 12.58

```
import java.io.*;
import java.util.*;
class Superclass
{
int num;
public Superclass(int num)
{
```

```
this.num = num;
}
void display()
{
System.out.println("super class: "+num);
}
}
class Subclass extends Superclass
{
int num;
public Subclass(int num,int Subnum)
{
super(num);
this.num = Subnum;
}
void display()
{
super.display();
System.out.println("sub class:"+num);
}
}
class rec
{
public static void main(String args[])
{
int superclassnum = 100;
int subclassnum = 200;
Subclass obj = new Subclass(superclassnum, subclassnum);
        obj.display();

}
}
```

18. Display Multiplication table for 5 and 10 using various stages of life cycle of the thread by generating a suitable code in Java.

    Sample Input 5, 10
    5 X 1 = 5
    5 X 2 =10
     ….
    10 X 1 =10
    10 X 2 = 20
    ….
    Test Cases:
    1. 10, 20
    2. -10, -30

3. 0, 0
4. SIX, SIX
5. 9.8, 9.6

```
import java.io.*;
import java.util.*;
class mul extends Thread
{
public void run()
{
int n;
Scanner s = new Scanner(System.in);
System.out.println("enter n:");
n = s.nextInt();
for(int i=1;i<=10;i++)
{
System.out.println(n+" * "+ i + " = "+(n * i));
}
}
}
class rec
{
public static void main(String args[])
{
mul z = new mul();
z.start();
}
}
```

19. Using the concepts of thread with implementing Runnable interface in Java to generate Fibonacci series.

Sample Input : 5

Sample Output : 0 1 1 2 3 …..

Test Cases

1. 7
2. -10
3. 0
4. EIGHT FIVE
5. 12.65

```
import java.io.*;
import java.util.*;
class A implements Runnable
{
public void run()
{
int n,firstterm=0,secondterm=1,i=1;
System.out.println("enter n value:");
Scanner s = new Scanner(System.in);
n = s.nextInt();
while(i<=n)
{
System.out.println(firstterm);
int nextterm = firstterm + secondterm;
firstterm = secondterm;
secondterm = nextterm;
i++;
}
}
}
class rec
{
public static void main(String args[])
{
A a = new A();
Thread t = new Thread(a);
t.start();
}
}
```

20. Generate a Java code to find the sum of N numbers using array and throw ArrayIndexOutOfBoundsException when the loop variable beyond the size N.

Sample Input : 5
1 2 3 4 5
Sample Output : 15

Test Cases

1. 4, 10
2. -10
3. 0
4. EIGHT SEVEN
5. 12.68

```
import java.util.Scanner;

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

        // Get input for the number of elements (N)
        System.out.print("Enter the number of elements (N): ");
        int N = scanner.nextInt();
        // Create an array to store N numbers
        int[] numbers = new int[N];

        // Get input for each element
        System.out.println("Enter " + N + " numbers:");
        for (int i = 0; i < N; i++) {
            System.out.print("Number " + (i + 1) + ": ");
            numbers[i] = scanner.nextInt();
        }

        // Calculate the sum of the numbers
        int sum = 0;
        for (int i = 0; i < N; i++) {
            try {
                // Attempt to access the array element
                sum += numbers[i];
            } catch (ArrayIndexOutOfBoundsException e) {
                // Handle the exception
                System.out.println("ArrayIndexOutOfBoundsException: Loop variable beyond
array size.");
                System.exit(1); // Exit the program
            }
        }

        // Display the sum of the numbers
        System.out.println("Sum of numbers: " + sum);
    }
}
```

21. Using the concepts of thread with implementing Runnable interface in Java to find whether a given number is prime or not.

Sample Input : 5
Sample Output : 5 is Prime

Sample Output : 15

Test Cases

1. 4
2. -10
3. 0
4. EIGHT SEVEN
5. 11.48

```
public class PrimeChecker implements Runnable {
    private int number;

    public PrimeChecker(int number) {
        this.number = number;
    }

    @Override
    public void run() {
        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
    }

    private boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
        // Example: Check if 17 and 20 are prime using threads
        Thread thread1 = new Thread(new PrimeChecker(17));
        Thread thread2 = new Thread(new PrimeChecker(20));

        thread1.start();
        thread2.start();
    }
}
```

22. Generate a Java code to handle Exceptions such as Arithmetic Exception, ArrayIndexOutOfBoundsException, NullPointerException using Multi-Catch Statements.

23. Generate a Java Code to Write and Read the string “Computer Science and Engineering” using FileWriter and FileReader Class.
24. Create a java program to construct the volume of Box using default constructor method.
25. Accept the string “Welcome to Saveetha university” from the user and perform the following operations by writing a suitable Java code.

i) Replace any word in the given String
ii) Find the length
iii) Uppercase Conversion

26. Create a HashTable to maintain a bank detail which includes Account number and Customer name. Let Account number be the key in the HashTable. Write a Java program to implement the following operations in the HashTable
    i) Add 3 records
    ii) Display the size of HashTable
    iii) Clear the HashTable

27. Create a employee record using map interface and do the following operations.
    i. Add object   iii. Remove specified object
    ii. isEmpty or not   iv. Clear

28. Create a simple generics class with type parameters for sorting values of different types.

29. Develop a Java code to insert the following elements, using ListIterator to append + symbol in each element and print them in reverse order. {C, A, E, B, D, F}.

30. Generate a Java code to perform simple arithmetic operations and to throw Arithmetic Exception for Division-by-Zero.

31. Write a Java program to create three threads in parallel and display the natural numbers in orders using sleep() method.

32. If n = 8, then array ‘a’ will have 7 elements in the range from 1 to 8. For example {1, 4, 5, 3, 7, 8, 6}. One number will be missing in ‘a’ (2 in this case). Write a source code to find out that missing number

33. Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call
    1 - method of parent class by object of parent class
    2 - method of child class by object of child class
    3 - method of parent class by object of child class

34. Write a Java program to create a class Student and create constructor which assigns the values for the student details such as student name, register number, and five subject marks. Calculate the total and average of five subject marks and display the marks and average.
35. Generate a code to Count the Number of Words, Character and Lines from the File using Stream I/O in Java.
36. Generate a code to non-negative integer’s num1 and num2 represented as strings; return the product of num1 and num2, also represented as a string.
37. Implement pow(x, n), which calculates x raised to the power n

Input: x = 2.00000, n = 10
Output: 1024.00000

38. Given an integer array nums, find the subarray with the largest sum, and return its sum.
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: The subarray [4,-1, 2, 1] has the largest sum 6.
39. Write a Java program which creates only one object. If user attempts to create second object, he should not be able to create it. (Using Exception Handling).
40. There is an exam room with n seats in a single row labeled from 0 to n - 1.When a student enters the room, they must sit in the seat that maximizes the distance to the closest person. If there are multiple such seats, they sit in the seat with the lowest number. If no one is in the room, then the student sits at seat number 0.Design a class that simulates the mentioned exam room. Implement the ExamRoom class: ExamRoom (int n) Initializes the object of the exam room with the number of the seats n. int seat () Returns the label of the seat at which the next student will set. Void leave (int p) indicates that the student sitting at seat p will leave the room. It is guaranteed that there will be a student sitting at seat p.
Input["ExamRoom", "seat", "seat", "seat", "seat", "leave", "seat"]
[[10], [], [], [], [], [4], []]
Output
[null, 0, 9, 4, 2, null, 5]

41. You have n tiles, where each tile has one letter tiles[i] printed on it. Return the number of possible non-empty sequences of letters you can make using the letters printed on those tiles.
Input: tiles = "AAB"
Output: 8
Explanation: The possible sequences are "A", "B", "AA", "AB", "BA", "AAB", "ABA", "BAA".
42. Write a program to read the data from the file and copy it on another file
43. Implementing FileReader and FileWriter class to read and write the data from file
44. Write a program to read the file using BufferedReader class
45. Write a program for counting number of characters, words and lines in a file
46. Write a program in Java to input an NxN matrix and display it row-wise and column-wise
47. Write a java program which creates an interface IterF1 having 2 methods add () and sub (). Create a class which overloads the given methods for addition and subtraction of two numbers respectively.
48. Write a Java program to calculate the rate of interest of the employee’s Provident Fund use the try and catch and finally block.

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

INSTITUTE OF PLACEMENT AND TRAINING

CSA09 –JAVA PROGRAMMING

49. Write a program to create a class MyThread in this class a constructor, call the base class constructor, using super and starts the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently
50. Create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.