Java Programming – CSA0988

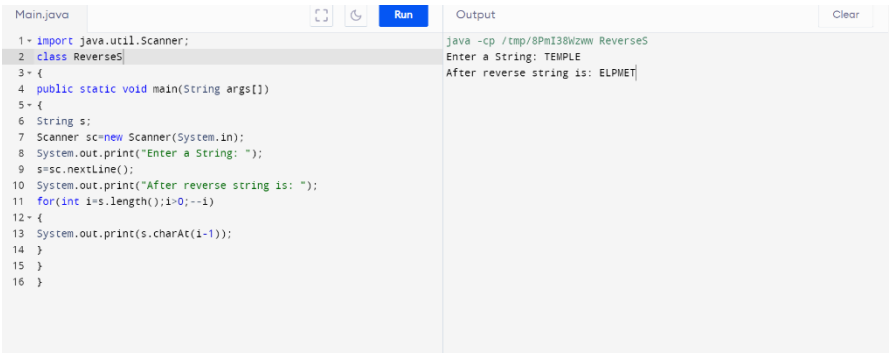
Shreya Rao SS

192020004

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

Code:

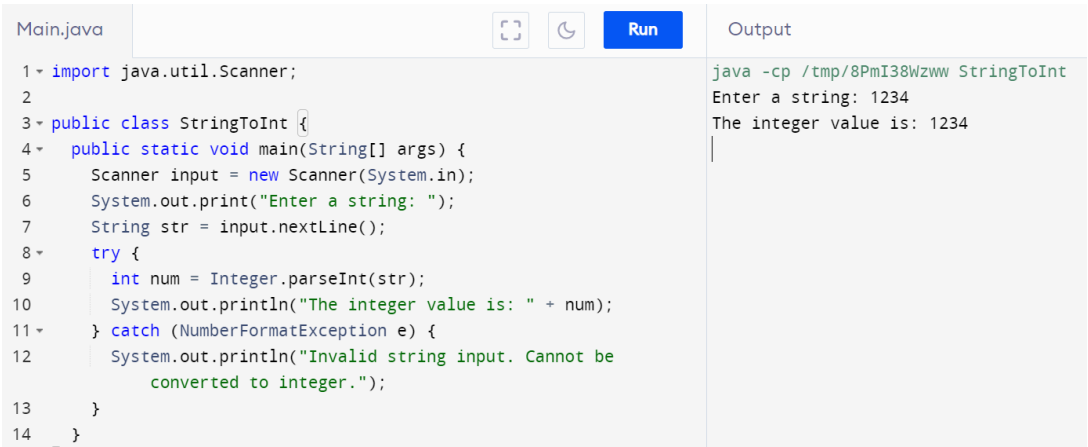
```
import java.util.Scanner;

class ReverseS

{

public static void main(String args[])

{

String s;

Scanner sc=new Scanner(System.in);

System.out.print("Enter a String: ");

s=sc.nextLine();

System.out.print("After reverse string is: ");

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

{

System.out.print(s.charAt(i-1));

}

}

}
```



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

Code:

```
import java.util.Scanner;

public class StringToInt {

  public static void main(String[] args) {

    Scanner input = new Scanner(System.in);

    System.out.print("Enter a string: ");

    String str = input.nextLine();

    try {

      int num = Integer.parseInt(str);

      System.out.println("The integer value is: " + num);

    } catch (NumberFormatException e) {

      System.out.println("Invalid string input. Cannot be converted to integer.");

    }

  }

}
```



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

Code:

```
import java.util.Scanner;

public class UserNameValidation {

  public static void main(String[] args) {

    Scanner input = new Scanner(System.in);

    System.out.print("Enter a user name: ");

    String username = input.nextLine();

    if (username.matches("^[a-zA-Z0-9]+$")) {

      System.out.println("Valid user name");

    } else {

      System.out.println("Invalid user name");

    }

  }

}
```

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:

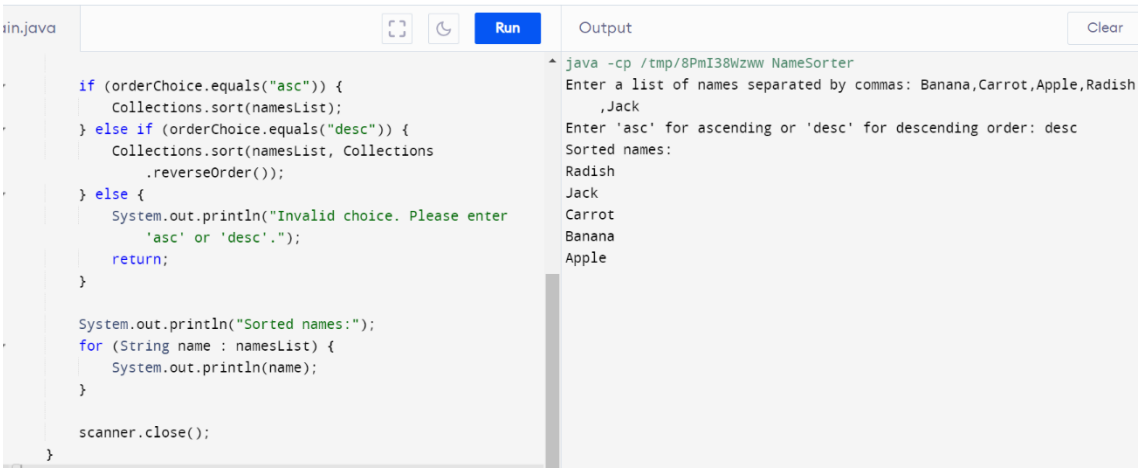
   Apple
   Banana

Carrot
Jack
Radish

Code:

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

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

        System.out.print("Enter a list of names separated by commas: ");
        String input = scanner.nextLine();
        String[] namesArray = input.split(",");
        List<String> namesList = new ArrayList<>();
        for (String name : namesArray) {
            namesList.add(name.trim());
        }

        System.out.print("Enter 'asc' for ascending or 'desc' for descending order: ");
        String orderChoice = scanner.nextLine();

        if (orderChoice.equals("asc")) {
            Collections.sort(namesList);
        } else if (orderChoice.equals("desc")) {
            Collections.sort(namesList, Collections.reverseOrder());
        } else {
            System.out.println("Invalid choice. Please enter 'asc' or 'desc'.");
            return;
```

```
        }

        System.out.println("Sorted names:");
        for (String name : namesList) {
            System.out.println(name);
        }

        scanner.close();
    }
}
```

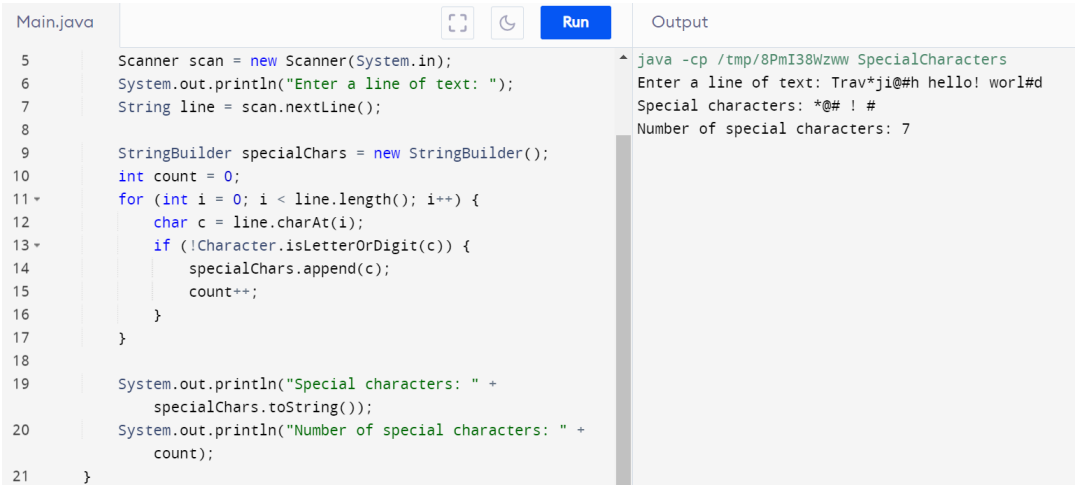


5. Write a program to print the special characters separately and print number of Special characters in the line?

Code:

```
import java.util.Scanner;

public class SpecialCharacters {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter a line of text: ");

        String line = scan.nextLine();

        StringBuilder specialChars = new StringBuilder();

        int count = 0;
```

```
        for (int i = 0; i < line.length(); i++) {
            char c = line.charAt(i);
            if (!Character.isLetterOrDigit(c)) {
                specialChars.append(c);
                count++;
            }
        }
        System.out.println("Special characters: " + specialChars.toString());
        System.out.println("Number of special characters: " + count);
    }
}
```

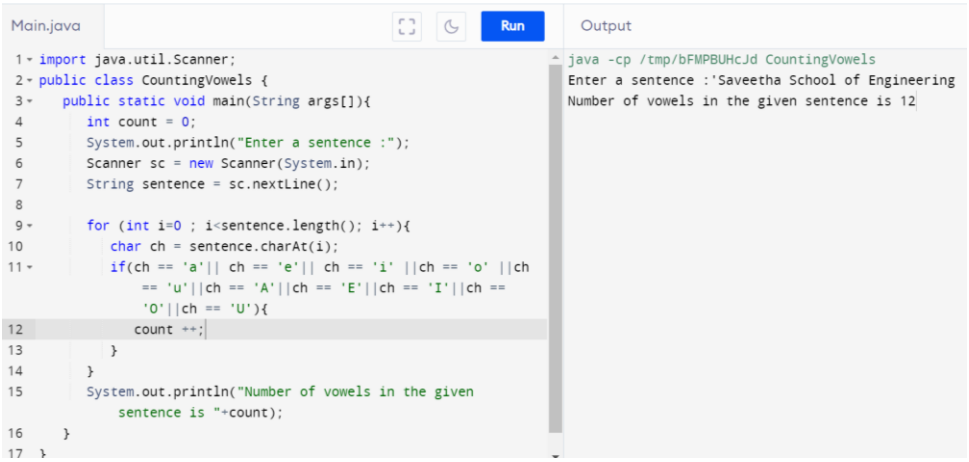


6. Write a program to print the number of vowels in the given statement?
Sample Input:
Saveetha School of Engineering
Sample Output:
Number o vowels = 12

Code:

```
import java.util.Scanner;

public class CountingVowels {

  public static void main(String args[]){

    int count = 0;

    System.out.println("Enter a sentence :");
```

```
    Scanner sc = new Scanner(System.in);
    String sentence = sc.nextLine();


    for (int i=0 ; i<sentence.length(); i++){
      char ch = sentence.charAt(i);
      if(ch == 'a'|| ch == 'e'|| ch == 'i' ||ch == 'o' ||ch == 'u'||ch == 'A'||ch == 'E'||ch == 'I'||ch == 'O'||ch == 'U'){
        count ++;
      }
    }
    System.out.println("Number of vowels in the given sentence is "+count);
  }
}
```

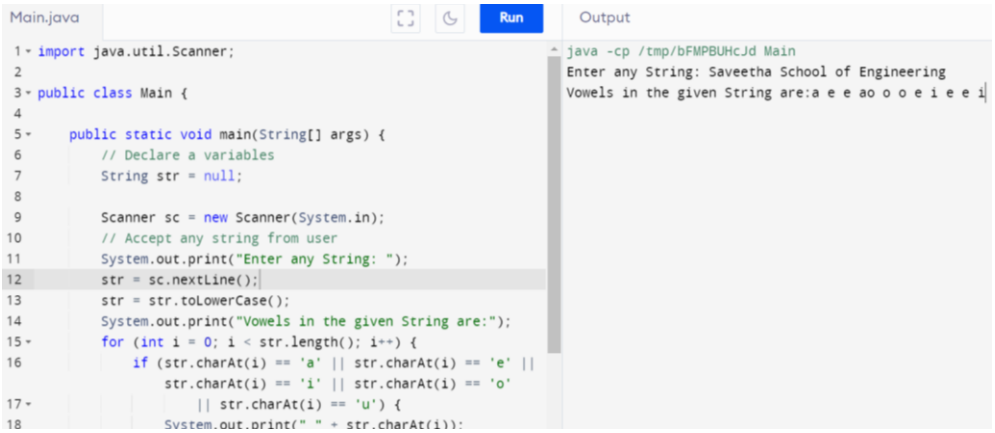


7. Write a program to print consonants and vowels separately in the given word
   Sample Input:
      Given Word: Engineering
   Sample Output:
      Consonants: n g n r n g
      Vowels: e i e ei

Code:

```
import java.util.Scanner;

public class Main {

        public static void main(String[] args) {

                String str = null;
```

```
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter any String: ");
        str = sc.nextLine();
        str = str.toLowerCase();
        System.out.print("Vowels in the given String are:");
        for (int i = 0; i < str.length(); i++) {
        if (str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o'
                                    || str.charAt(i) == 'u') {
                            System.out.print(" " + str.charAt(i));
                }
        }

    }
```



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

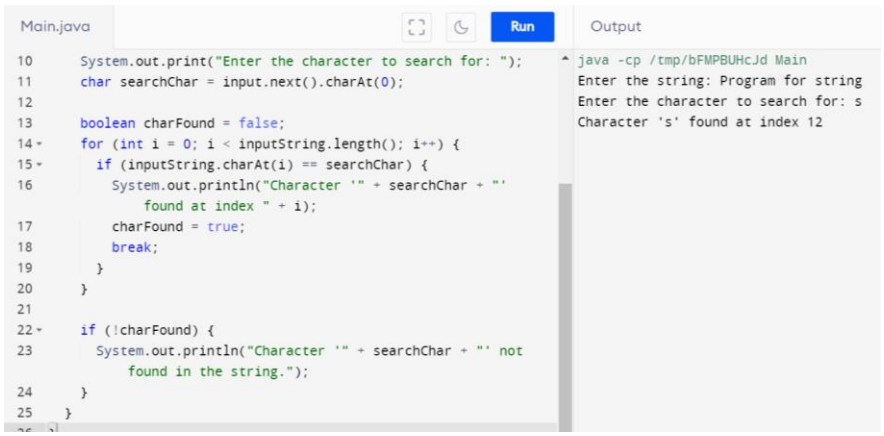
Sample Output:

P is found in string at index: 8

Code:

```
import java.util.Scanner;

public class Main {

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

```
    Scanner input = new Scanner(System.in);
    System.out.print("Enter the string: ");
    String inputString = input.nextLine();
    System.out.print("Enter the character to search for: ");
    char searchChar = input.next().charAt(0);
    boolean charFound = false;
    for (int i = 0; i < inputString.length(); i++) {
      if (inputString.charAt(i) == searchChar) {
        System.out.println("Character '" + searchChar + "' found at index " + i);
        charFound = true;
        break;
      }
    }
    if (!charFound) {
      System.out.println("Character '" + searchChar + "' not found in the string.");
    }
  }
}
```



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

Sample Input:

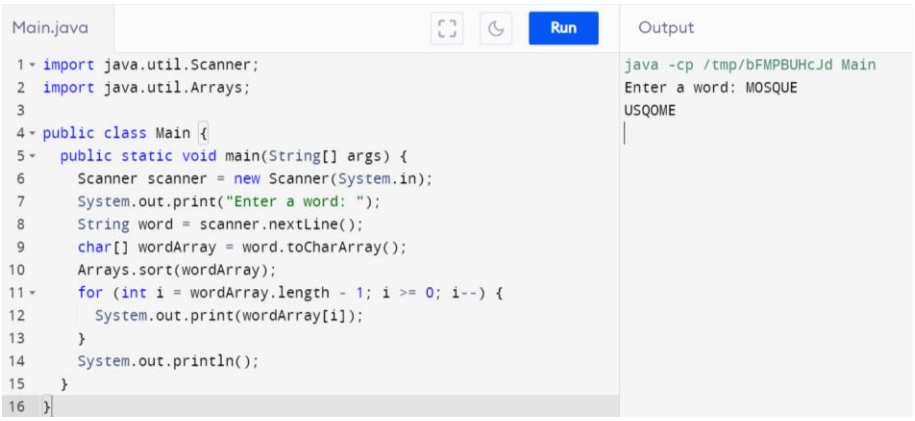
Enter the word: MOSQUE

Sample Output:

Alphabetical Order: U S Q O M E

Code:

```
import java.util.Scanner;
import java.util.Arrays;
public class Main {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a word: ");
    String word = scanner.nextLine();
    char[] wordArray = word.toCharArray();
    Arrays.sort(wordArray);
    for (int i = wordArray.length - 1; i >= 0; i--) {
      System.out.print(wordArray[i]);
    }
    System.out.println();
  }
}
```



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

Sample Input & Output:

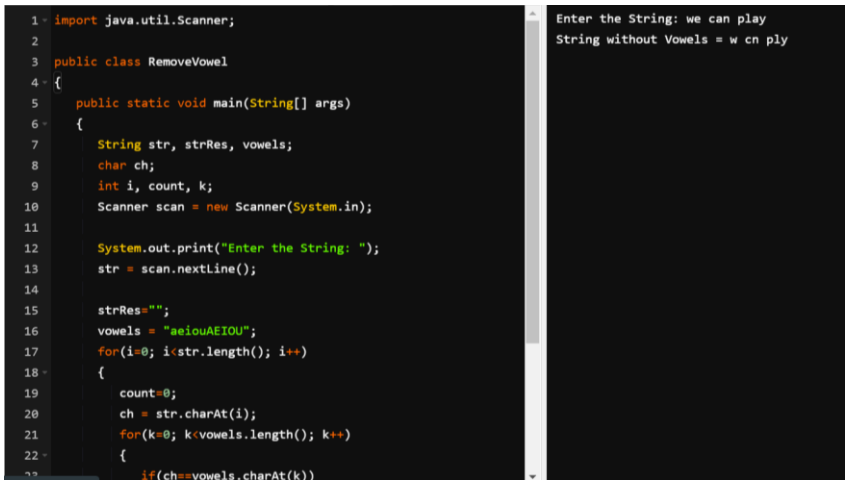
Enter a string: we can play the game

The string without vowels is: w cn ply thgm

Code:

```
import java.util.Scanner;

public class RemoveVowel

{

  public static void main(String[] args)

  {

    String str, strRes, vowels;

    char ch;

    int i, count, k;

    Scanner scan = new Scanner(System.in);

    System.out.print("Enter the String: ");

    str = scan.nextLine();

    strRes="";

    vowels = "aeiouAEIOU";

    for(i=0; i<str.length(); i++)

    {

      count=0;

      ch = str.charAt(i);

      for(k=0; k<vowels.length(); k++)

      {

        if(ch==vowels.charAt(k))

          count++;

      }

      if(count==0)

        strRes = strRes + ch;

    }

    System.out.println("\nString without Vowels = " +strRes);

  }
```

}



```
import java.util.Scanner;

public class RemoveVowel
{
   public static void main(String[] args)
   {
      String str, strRes, vowels;
      char ch;
      int i, count, k;
      Scanner scan = new Scanner(System.in);

      System.out.print("Enter the String: ");
      str = scan.nextLine();

      strRes="";
      vowels = "aeiouAEIOU";
      for(i=0; i<str.length(); i++)
      {
         count=0;
         ch = str.charAt(i);
         for(k=0; k<vowels.length(); k++)
         {
            if(ch==vowels.charAt(k))
```

```
Enter the String: we can play
String without Vowels = w cn ply
```

11. Write a program for matrix multiplication?

Sample Input:

Mat1 = 1 2
5 3

Mat2 = 2 3
4 1

Sample Output:

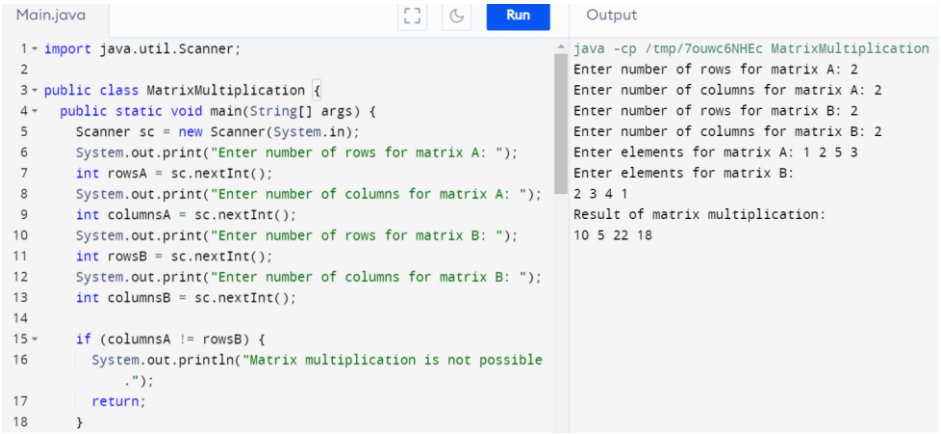
Mat Sum = 10 5
22 18

Code:

```
import java.util.Scanner;

public class MatrixMultiplication {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.print("Enter number of rows for matrix A: ");

    int rowsA = sc.nextInt();

    System.out.print("Enter number of columns for matrix A: ");

    int columnsA = sc.nextInt();

    System.out.print("Enter number of rows for matrix B: ");

    int rowsB = sc.nextInt();

    System.out.print("Enter number of columns for matrix B: ");

    int columnsB = sc.nextInt();
```

```
if (columnsA != rowsB) {
  System.out.println("Matrix multiplication is not possible.");
  return;
}
int[][] matrixA = new int[rowsA][columnsA];
int[][] matrixB = new int[rowsB][columnsB];
int[][] result = new int[rowsA][columnsB];
System.out.println("Enter elements for matrix A: ");
for (int i = 0; i < rowsA; i++) {
  for (int j = 0; j < columnsA; j++) {
    matrixA[i][j] = sc.nextInt();
  }
}
System.out.println("Enter elements for matrix B: ");
for (int i = 0; i < rowsB; i++) {
  for (int j = 0; j < columnsB; j++) {
    matrixB[i][j] = sc.nextInt();
  }
}
for (int i = 0; i < rowsA; i++) {
  for (int j = 0; j < columnsB; j++) {
    for (int k = 0; k < columnsA; k++) {
      result[i][j] += matrixA[i][k] * matrixB[k][j];
    }
  }
}
System.out.println("Result of matrix multiplication: ");
for (int i = 0; i < rowsA; i++) {
  for (int j = 0; j < columnsB; j++) {
    System.out.print(result[i][j] + " ");
  }
```

```
    System.out.println();
  }
 }
}
```



12. Write a program for matrix addition?

Sample Input:

Mat1 = 1 2
5 3

Mat2 = 2 3
4 1

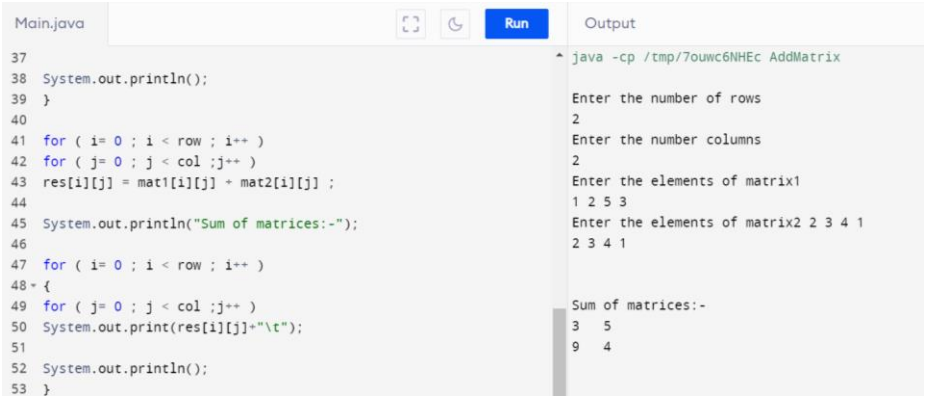
Sample Output:

Mat Sum = 3 5
9 4

Code:

```
import java.util.Scanner;

class AddMatrix

{

public static void main(String args[])

{

int row, col,i,j;

Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows");

row = in.nextInt();

System.out.println("Enter the number columns");

col = in.nextInt();

int mat1[][] = new int[row][col];
```

```
int mat2[][] = new int[row][col];
int res[][] = new int[row][col];
System.out.println("Enter the elements of matrix1");
for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
mat1[i][j] = in.nextInt();
System.out.println();
}
System.out.println("Enter the elements of matrix2");
for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
mat2[i][j] = in.nextInt();
System.out.println();
}
for ( i= 0 ; i < row ; i++ )
for ( j= 0 ; j < col ;j++ )
res[i][j] = mat1[i][j] + mat2[i][j] ;
System.out.println("Sum of matrices:-");
for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
System.out.print(res[i][j]+"\t");
System.out.println();
}
}
}
```

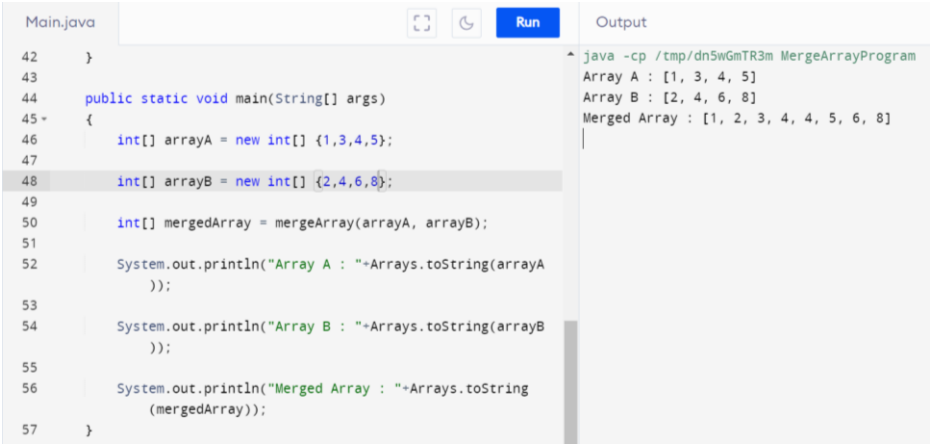


13. Write a program for Merge two sorted arrays using Array list
Input: arr1[] = { 1, 3, 4, 5}, arr2[] = {2, 4, 6, 8}
Output: arr3[] = {1, 2, 3, 4, 4, 5, 6, 8}

Code:

```
import java.util.Arrays;

public class MergeArrayProgram

{

   private static int[] mergeArray(int[] arrayA, int[] arrayB)

   {

      int[] mergedArray = new int[arrayA.length + arrayB.length];

      int i=0, j=0, k=0;

      while (i < arrayA.length && j < arrayB.length)

      {

         if (arrayA[i] < arrayB[j])

         {

            mergedArray[k] = arrayA[i];

            i++;

            k++;

         }

         else

         {

            mergedArray[k] = arrayB[j];

            j++;

            k++;

         }
```

```
        }
        while (i < arrayA.length)
        {
            mergedArray[k] = arrayA[i];
            i++;
            k++;
        }
        while (j < arrayB.length)
        {
            mergedArray[k] = arrayB[j];
            j++;
            k++;
        }
        return mergedArray;
    }
    public static void main(String[] args)
    {
        int[] arrayA = new int[] {1,3,4,5};
        int[] arrayB = new int[] {2,4,6,8};
        int[] mergedArray = mergeArray(arrayA, arrayB);
        System.out.println("Array A : "+Arrays.toString(arrayA));
        System.out.println("Array B : "+Arrays.toString(arrayB));
        System.out.println("Merged Array : "+Arrays.toString(mergedArray));
    }
}
```

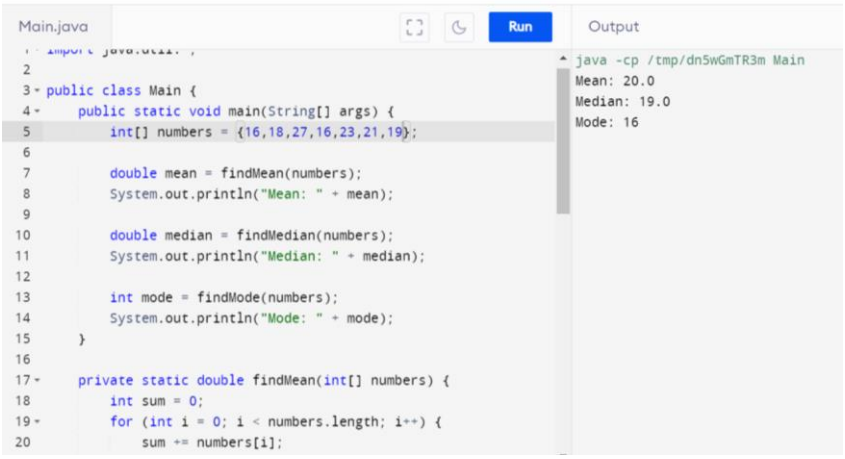


14. Find the Mean, Median, Mode of the array of numbers?
Sample Input;:
Array of elements = {16, 18, 27, 16, 23, 21, 19}
Sample Output:
Mean = 20
Median = 19
Mode = 16

Code:

```
import java.util.*;

public class Main {

   public static void main(String[] args) {

      int[] numbers = {16,18,27,16,23,21,19};

      double mean = findMean(numbers);

      System.out.println("Mean: " + mean);

      double median = findMedian(numbers);

      System.out.println("Median: " + median);

      int mode = findMode(numbers);

      System.out.println("Mode: " + mode);

   }

   private static double findMean(int[] numbers) {

      int sum = 0;

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

         sum += numbers[i];

      }

      return (double) sum / numbers.length;

   }
```

```
    private static double findMedian(int[] numbers) {
        Arrays.sort(numbers);
        if (numbers.length % 2 == 0) {
            return (double) (numbers[numbers.length / 2] + numbers[numbers.length / 2 - 1]) / 2;
        } else {
            return (double) numbers[numbers.length / 2];
        }
    }
    private static int findMode(int[] numbers) {
        HashMap<Integer, Integer> frequency = new HashMap<>();
        int maxValue = 0;
        int mode = -1;
        for (int i = 0; i < numbers.length; i++) {
            if (frequency.containsKey(numbers[i])) {
                frequency.put(numbers[i], frequency.get(numbers[i]) + 1);
            } else {
                frequency.put(numbers[i], 1);
            }
            if (frequency.get(numbers[i]) > maxValue) {
                maxValue = frequency.get(numbers[i]);
                mode = numbers[i];
            }
        }
        return mode;
    }
}
```



15. Write a program to print Right Triangle Star Pattern

Sample Input:: n = 5

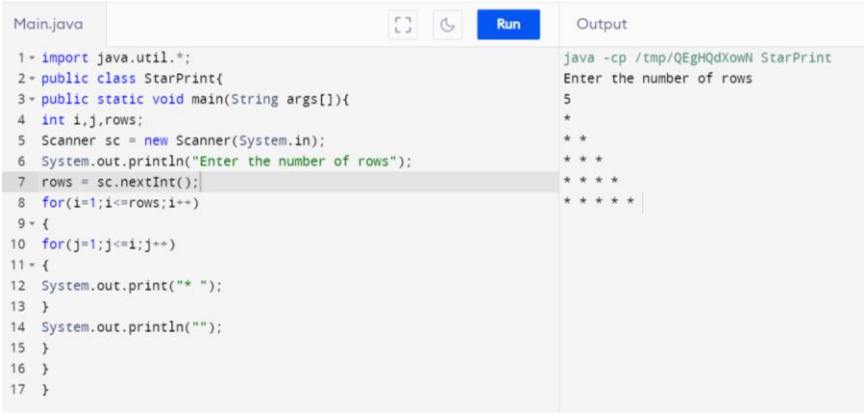
Output:

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

Code:

```
import java.util.*;

public class StarPrint{

public static void main(String args[]){

int i,j,rows;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rows");

rows = sc.nextInt();

for(i=1;i<=rows;i++)

{

for(j=1;j<=i;j++)

{

System.out.print("* ");

}

System.out.println("");
```

```
}
}
}}
System.out.println("");
}
}
}
```



16. Write a program to print the below pattern?

```
                1
            1       1
        1       2       1
    1       3       3       1
1       4       6       4       1
```

Code:

```
import java.util.Scanner;
public class MainClass
{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter number of rows: ");
      int noOfRows = sc.nextInt();
      int rowCount = 1;
```

```
        System.out.println("Here Is Your Pyramid");
        for (int i = noOfRows; i > 0; i--)
        {
            for (int j = 1; j <= i*2; j++)
            {
                System.out.print(" ");
            }
            for (int j = 1; j <= rowCount; j++)
            {
                System.out.print(j+" ");
            }
            for (int j = rowCount-1; j >= 1; j--)
            {
                System.out.print(j+" ");
            }
            System.out.println();
            rowCount++;
        }
    }
```

```
Enter number of rows: 5
Here Is Your Pyramid
          1
        1 2 1
      1 2 3 2 1
    1 2 3 4 3 2 1
  1 2 3 4 5 4 3 2 1
```

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

Code:

```
import java.util.Scanner;

public class RectangleStar {

        private static Scanner sc;

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

```
        {
                int rows, columns, i, j;
                sc = new Scanner(System.in);

                System.out.print(" Please Enter Number of Rows : ");
                rows = sc.nextInt();

                System.out.print(" Please Enter Number of Columns : ");
                columns = sc.nextInt();

                for(i = 1; i <= rows; i++)
                {
                        for(j = 1; j <= columns; j++)
                        {
                                System.out.print("* ");
                        }
                        System.out.print("\n");
                }
        }
}
```

```
Please Enter Number of Rows : 3
Please Enter Number of Columns : 5
* * * * *
* * * * *
* * * * *
```

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

Code:

```
import java.util.Scanner;

public class InvPyr

{

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

```
{

        Scanner sc=new Scanner(System.in);
        System.out.println("Enter N : ");
        int n=sc.nextInt();
        System.out.print("Enter Symbol : ");

      char c = sc.next().charAt(0);

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

              {
                    System.out.print(" ");
              }
              for(int j=0;j<(i*2)-1;j++)

              {
                    System.out.print(c);
              }
             System.out.println();
        }
}
```

```
Enter N :
5
Enter Symbol : *
*********
 *******
  *****
   ***
    *
```

19. Write a program to print the following pattern

Sample Input:

Enter the Character to be printed: %

Max Number of time printed: 3

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

Code:

```
import java.util.Scanner;

public class CharNumberPattern {

   public static void main(String args[]) {

      Scanner scanner = new Scanner(System.in);


      System.out.println("Enter the character to be printed: ");

      char ch = scanner.next().charAt(0);


      System.out.println("Max number of times to be printed '" + ch + "' : ");

      int num = scanner.nextInt();


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

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

            System.out.print(ch);

         }

         System.out.println();

      }

   }
```

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

Sample Input:

Number: 14567

Sample Output:

Reverse Number: 76541

```
public class ReverseNumber
{
public static void main(String[] args)
{
int number = 14567, reverse = 0;
while(number != 0)
{
int remainder = number % 10;
reverse = reverse * 10 + remainder;
number = number/10;
}
System.out.println("The reverse of the given number is: " + reverse);
}
}
```

```
perl

The reverse of the given number is: 76541
```

21. 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

```
import java.util.Scanner;

public class Voting {
```

```
 public static void main(String[] args)
{
   int age, diff;
   Scanner scan = new Scanner(System.in);
   System.out.println("Please enter your age: ");
   age = scan.nextInt();
   if(age>=18)
   {
      System.out.println("You are eligible for voting.");
   }
   else
   {
     diff = (18 - age);
   System.out.println("You can vote after: "+ diff + " years");
   }
 }
}
```

```yaml
Please enter your age:
16
You can vote after: 2 years
```

22. 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

```
import java.util.Scanner;
public class PrintLcmHcf {
   public static void main(String[] args) {
```

```
        int a, b, t, aTemp, bTemp, lcm, gcd;
        Scanner scanner;
        scanner = new Scanner(System.in);
        System.out.println("Enter Two Number");
        a = scanner.nextInt();
        b = scanner.nextInt();
        aTemp = a;
        bTemp = b;
        while (bTemp != 0) {
            t = bTemp;
            bTemp = aTemp % bTemp;
            aTemp = t;
        }
        gcd = aTemp;
        lcm = (a * b) / gcd;
        System.out.println("LCM = " + lcm);
        System.out.println("GCD = " + gcd);
    }
}
```

```
makefile

LCM = 80
GCD = 4
```

23. Write a program to print the Fibonacci series.

Sample Input:

Enter the n value: 6

```
import java.util.Scanner;

public class Fibonacci

{

   public static void main(String[] args)

   {
```

```
        int n, a = 0, b = 0, c = 1;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter value of n:");
        n = s.nextInt();
        System.out.print("Fibonacci Series:");
        for(int i = 1; i <= n; i++)
        {
           a = b;
           b = c;
           c = a + b;
           System.out.print(a+" ");
        }
    }
}
```

```
mathematica
Fibonacci Series:0 0 1 1 2 3
```

24. Write a program to print all the composite numbers between a and b?
Sample Input:
A = 12
B = 19

```
import java.util.Scanner;

public class CompositeNumbers {

    static boolean isComposite(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 main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter value of a: ");
      int a = scanner.nextInt();
      System.out.print("Enter value of b: ");
      int b = scanner.nextInt();
      System.out.println("Composite Numbers between " + a + " and " + b + ":");
      for (int i = a; i <= b; i++) {
         if (isComposite(i)) {
            System.out.print(i + " ");
         }
      }
      scanner.close();
   }
}
```

```
Composite Numbers between 10 and 30:
10 12 14 15 16 18 20 21 22 24 25 26 27 28 30
```

25. Find the factorial of n?

Sample Input:

N = 4

Sample Output:

4 Factorial = 24

```
class Factorial{
 public static void main(String args[]){
  int i,fact=1;
  int number=5;//It is the number to calculate factorial
```

```
 for(i=1;i<=number;i++){
    fact=fact*i;
 }
 System.out.println("Factorial of "+number+" is: "+fact);
}
}
```

```
Factorial of 4 is: 24
```

26. 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

```
import java.util.Scanner;
public class PrimeCompositeCounter {
   public static boolean isPrime(int num) {
      if (num <= 1) {
         return false;
      }
      for (int i = 2; i * i <= num; i++) {
         if (num % i == 0) {
            return false;
         }
      }
      return true;
   }

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

      System.out.print("Enter the number of elements: ");
      int n = scanner.nextInt();

      int primeCount = 0;
```

```
        int compositeCount = 0;

        System.out.println("Enter the elements:");

        for (int i = 0; i < n; i++) {
            int num = scanner.nextInt();
            if (isPrime(num)) {
                primeCount++;
            } else {
                compositeCount++;
            }
        }

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

        scanner.close();
    }
}
```

```
Number of prime numbers: 5
Number of composite numbers: 3
```

27. Program to print hollow square

```
import java.util.Scanner;

public class Pattern1 {

    public static void main(String[] args) {

            Scanner sc = new Scanner(System.in);

            System.out.println("Enter row and col">);

            int row = sc.nextInt();

            int col = sc.nextInt();

            for (int i = 1; i <=row; i++) {

                    for (int j = 1; j <= col; j++)

                            if((i==1 || i==col) || (j==1 || j==col))

                                    System.out.print("*");

                            else

                                    System.out.print(" ");


                    System.out.println();
```

```
        }
    }
}
```

```
Enter row and col
4 4
****
*  *
*  *
****
```

28. 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

```
import java.util.Scanner;

public class BonusCalculator {

   public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter the grade of the employee: ");

      char grade = scanner.next().charAt(0);

      System.out.print("Enter the employee salary: ");

      double salary = scanner.nextDouble();

      double bonus = 0.0;

      if (grade == 'B') {

         bonus = 0.02 * salary;

      }

      double totalSalary = salary + bonus;

      System.out.println("Salary = " + salary);

      System.out.println("Bonus = " + bonus);
```

```
        System.out.println("Total to be paid: " + totalSalary);
        scanner.close();
    }
}
```

```
Enter the grade of the employee: A
Enter the employee salary: 5000
Salary = 5000.0
Bonus = 0.0
Total to be paid: 5000.0
```

29. 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)**

```
public class PerfectNumbers {
    public static boolean isPerfect(int num) {
        int sum = 1;
        for (int i = 2; i * i <= num; i++) {
            if (num % i == 0) {
                sum += i;
                if (i != num / i) {
                    sum += num / i;
                }
            }
        }
        return sum == num && num != 1;
    }
    public static void main(String[] args) {
        int n = 5; // Number of perfect numbers to find
        int count = 0;
        int num = 2; // Starting number to check for perfection
        System.out.println("First " + n + " perfect numbers:");
```

```
        while (count < n) {
            if (isPerfect(num)) {
                System.out.print(num + " ");
                count++;
            }
            num++;
        }
    }
}
```

```
First 5 perfect numbers:
6 28 496 8128 130816
```

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

```
class Main
{
  static int removeDuplicates(int arr[], int n)
  {
    if (n==0 || n==1)
      return n;
int j = 0;
for (int i=0; i<n-1; i++)
if (arr[i] != arr[i+1])
arr[j++] = arr[i];
arr[j++] = arr[n-1];
return j;
}
public static void main (String[] args)
{
int arr[] = {10, 20, 20, 30, 40, 40, 50};
int n = arr.length;
```

```
n = removeDuplicates(arr, n);

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

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

}

}
```

```
10 20 30 40 50
```

31. 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

```
import java.util.Scanner;
class Test
{
public static void main(String args[])
{
   int days, years, weeks;
   Scanner op=new Scanner(System.in);
   /* Input total number of days from user */
   System.out.print("Enter days: ");
   days=op.nextInt();
   years = (days / 365);   // Ignoring leap year
   weeks = (days % 365) / 7;
   days  = days - ((years * 365) + (weeks * 7));
   System.out.println("YEARS: "+years);
   System.out.println("WEEKS: "+weeks);
   System.out.println("DAYS:  "+days);
}
}
```

```
Enter days: 458

YEARS: 1
WEEKS: 19
DAYS:  5
```

32. 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.

```
import java.util.Scanner;

class CollegeUserCounter {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of student users: ");

        int studentUsers = scanner.nextInt();

        System.out.print("Enter the number of staff users: ");

        int staffUsers = scanner.nextInt();

        int nonTeachingStaffUsers = staffUsers / 3;

        int totalUsers = studentUsers + staffUsers + nonTeachingStaffUsers;

        System.out.println("Total student users: " + studentUsers);

        System.out.println("Total staff users: " + staffUsers);

        System.out.println("Total non-teaching staff users: " + nonTeachingStaffUsers);

        System.out.println("Total users in the college: " + totalUsers);

        scanner.close();

    }

}
```

```
Enter the number of student users: 856
Enter the number of staff users: 126
Total student users: 856
Total staff users: 126
Total non-teaching staff users: 42
Total users in the college: 1024
```

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

```
import java.util.HashSet;

import java.util.Scanner;

public class UniquePermutations {

    static HashSet<String> uniquePermutations = new HashSet<>();
```

```
public static void permute(String str, int l, int r) {
    if (l == r) {
        uniquePermutations.add(str);
    } else {
        for (int i = l; i <= r; i++) {
            str = swap(str, l, i);
            permute(str, l + 1, r);
            str = swap(str, l, i);
        }
    }
}
public static String swap(String a, int i, int j) {
    char temp;
    char[] charArray = a.toCharArray();
    temp = charArray[i];
    charArray[i] = charArray[j];
    charArray[j] = temp;
    return String.valueOf(charArray);
}
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a number: ");
    int number = scanner.nextInt();
    String numStr = String.valueOf(number);
    permute(numStr, 0, numStr.length() - 1);
    System.out.println("Unique permutations of " + number + ":");
    for (String permutation : uniquePermutations) {
        System.out.println(permutation);
    }
    scanner.close();
}
```

}

```
Enter a number: 143
Unique permutations of 143:
413
431
314
341
134
143
```

34. 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

```
import java.util.Scanner;

class BankAccount {

    private String accountNumber;

    private String depositorName;

    private String accountType;

    private double balance;

    public BankAccount(String accountNumber, String depositorName, String accountType) {

        this.accountNumber = accountNumber;

        this.depositorName = depositorName;

        this.accountType = accountType;

        this.balance = 10000.00; // Initial balance of Rs. 10000

    }
```

```
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited Rs. " + amount);
        displayBalance();
    }

    public void withdraw(double amount) {
        if (balance - amount >= 500.00) {
            balance -= amount;
            System.out.println("Withdrawn Rs. " + amount);
            displayBalance();
        } else {
            System.out.println("Insufficient balance. Minimum balance must be maintained.");
        }
    }
    public void displayBalance() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Depositor Name: " + depositorName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: Rs. " + balance);
    }
}
public class BankAccountDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Account Number: ");
        String accountNumber = scanner.nextLine();
        System.out.print("Enter Depositor Name: ");
        String depositorName = scanner.nextLine();
        System.out.print("Enter Account Type (Savings/Current): ");
```

```
        String accountType = scanner.nextLine();

        BankAccount account = new BankAccount(accountNumber, depositorName, accountType);
        account.displayBalance();
        System.out.print("Enter amount to deposit: ");
        double depositAmount = scanner.nextDouble();
        account.deposit(depositAmount);

        System.out.print("Enter amount to withdraw: ");
        double withdrawAmount = scanner.nextDouble();
        account.withdraw(withdrawAmount);
        scanner.close();
    }
}
```

```
Account Number: 100
Depositor Name: Raja
Account Type: Savings
Balance: Rs. 18000.0
Enter amount to withdraw: 12000
Withdrawn Rs. 12000.0
Account Number: 100
Depositor Name: Raja
Account Type: Savings
Balance: Rs. 6000.0
```

35. Hollow square with $ pattern

```
public class SquareDollarPattern
{
    public static void main(String[] args)
    {
        int size = 5;
        for (int i = 0; i < size; i++)
        {
            for (int j = 0; j < size; j++)
```

```
        {
            if (i == 0 || i == size - 1 ||
                j == 0 || j == size - 1)
                System.out.print("$");
            else
                System.out.print(" ");
        }
        System.out.println();
    }
  }
}
```

```
$$$$$
$   $
$   $
$   $
$$$$$
```

36. 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)

```
import java.util.Scanner;

public class ReverseAndAddPalindrome {

    public static int reverseNumber(int num) {
        int reverse = 0;
        while (num > 0) {
            int digit = num % 10;
            reverse = reverse * 10 + digit;
            num /= 10;
        }
        return reverse;
```

```
    }
    public static boolean isPalindrome(int num) {
        return num == reverseNumber(num);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int input = scanner.nextInt();
        int reverse, sum;
        int iterations = 0;
        while (true) {
            reverse = reverseNumber(input);
            sum = input + reverse;
            System.out.println(input + " + " + reverse + " = " + sum);
            if (isPalindrome(sum)) {
                System.out.println("Palindrome obtained: " + sum);
                break;
            }
            input = sum;
            iterations++;
        }
        System.out.println("Total iterations: " + iterations);
        scanner.close();
    }
}
```

```
Enter a number: 7325
7325 + 5237 = 12562
12562 + 26521 = 39083
39083 + 38093 = 77176
77176 + 67177 = 144353
Palindrome obtained: 497794
Total iterations: 4
```

37. 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

```
class Customer {
    private int accountNo;
    private String accName;
    private int balance = 10000; // Existing bank balance
    public Customer(int accountNo, String accName) {
        this.accountNo = accountNo;
        this.accName = accName;
    }
    public synchronized void deposit(int amount) {
        balance += amount;
        System.out.println("Deposited: " + amount + ", New Balance: " + balance);
        notify(); // Notify waiting threads
    }
    public synchronized void withdraw(int amount) {
        if (balance < amount) {
            System.out.println("Insufficient balance. Waiting for deposit...");
            try {
                wait(); // Wait for deposit
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
```

```
        balance -= amount;
        System.out.println("Withdrawal success. Remaining balance: " + balance);
    }
}
public class Main {
    public static void main(String[] args) {
        Customer customer = new Customer(123, "John");
        Thread depositThread = new Thread(() -> customer.deposit(12000));
        Thread withdrawThread = new Thread(() -> customer.withdraw(3000));
        depositThread.start();
        withdrawThread.start();
        try {
            depositThread.join();
            withdrawThread.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

```
Deposited: 12000, New Balance: 22000
Withdrawal success. Remaining balance: 19000
```

38. 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

```
import java.util.HashMap;

public class RomanToInteger {

    public static int romanToInt(String s) {

        HashMap<Character, Integer> romanValues = new HashMap<>();

        romanValues.put('I', 1);

        romanValues.put('V', 5);

        romanValues.put('X', 10);

        romanValues.put('L', 50);

        romanValues.put('C', 100);

        romanValues.put('D', 500);

        romanValues.put('M', 1000);

        int result = 0;

        int prevValue = 0;

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

            int currValue = romanValues.get(s.charAt(i));

            if (currValue >= prevValue) {

                result += currValue;

            } else {

                result -= currValue;

            }

            prevValue = currValue;

        }

        return result;

    }
```

```
    public static void main(String[] args) {
        String romanNumeral = "XXVII";
        int intValue = romanToInt(romanNumeral);
        System.out.println("Roman numeral: " + romanNumeral);
        System.out.println("Integer value: " + intValue);
    }
}
```

```
Roman numeral: XXVII
Integer value: 27
```

39. 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

```
import java.util.Scanner;
public class StudentGrades {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the marks in python: ");
        int S1 = scanner.nextInt();
        System.out.print("Enter the marks in c programming: ");
        int S2 = scanner.nextInt();

        System.out.print("Enter the marks in Mathematics: ");
        int S3 = scanner.nextInt();

        System.out.print("Enter the marks in Physics: ");
```

```
        int S4 = scanner.nextInt();
        int total = S1 + S2 + S3 + S4;
        double aggregate = total / 4.0;
        System.out.println("Total= " + total);
        System.out.println("Aggregate = " + aggregate);
        if (aggregate > 75) {
            System.out.println("DISTINCTION");
        } else if (aggregate >= 60 && aggregate < 75) {
            System.out.println("First Division");
        } else if (aggregate >= 50 && aggregate < 60) {
            System.out.println("Second Division");
        } else if (aggregate >= 40 && aggregate < 50) {
            System.out.println("Third Division");
        } else {
            System.out.println("Fail");
        }
        scanner.close();
    }
}
```

```
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
```

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

```
import java.util.Scanner;
public class AverageOfNumbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int positiveSum = 0;
        int positiveCount = 0;
        int negativeSum = 0;
        int negativeCount = 0;

        System.out.println("Enter numbers (-1 to stop): ");

        while (true) {
            int num = scanner.nextInt();

            if (num == -1) {
                break;
```

```
            }

            if (num > 0) {
                positiveSum += num;
                positiveCount++;
            } else if (num < 0) {
                negativeSum += num;
                negativeCount++;
            }
        }

        if (positiveCount > 0) {
            double positiveAverage = (double) positiveSum / positiveCount;
            System.out.println("Average of positive numbers: " + positiveAverage);
        } else {
            System.out.println("No positive numbers entered.");
        }

        if (negativeCount > 0) {
            double negativeAverage = (double) negativeSum / negativeCount;
            System.out.println("Average of negative numbers: " + negativeAverage);
        } else {
            System.out.println("No negative numbers entered.");
        }

        scanner.close();
    }
}
```

```
Enter numbers (-1 to stop):
10
-5
7
-3
-1
```

```
Average of positive numbers: 9.0
Average of negative numbers: -4.333333333333333
```

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

```
import java.util.Scanner;

public class CharCounter {

    public static void main(String[] args) {

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

```
        int lowerCaseCount = 0;
        int upperCaseCount = 0;
        int numberCount = 0;
        char ch;
        System.out.println("Enter * to exit...");
        do {
            System.out.print("Enter any character: ");
            ch = scanner.next().charAt(0);
            if (Character.isLowerCase(ch)) {
                lowerCaseCount++;
            } else if (Character.isUpperCase(ch)) {
                upperCaseCount++;
            } else if (Character.isDigit(ch)) {
                numberCount++;
            }
        } while (ch != '*');
        System.out.println("Total count of lower case: " + lowerCaseCount);
        System.out.println("Total count of upper case: " + upperCaseCount);
        System.out.println("Total count of numbers: " + numberCount);
        scanner.close();
    }
}
```

```
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: *
Total count of lower case: 2
Total count of upper case: 4
Total count of numbers: 0
```

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

```
public class Factorial {
   public static void main(String[] args) {
      int num = 6;
      long factorial = multiplyNumbers(num);
      System.out.println("Factorial of " + num + " = " + factorial);
   }
   public static long multiplyNumbers(int num)
   {
      if (num >= 1)
         return num * multiplyNumbers(num - 1);
      else
         return 1;
   }
}
```

```
Factorial of 6 = 720
```

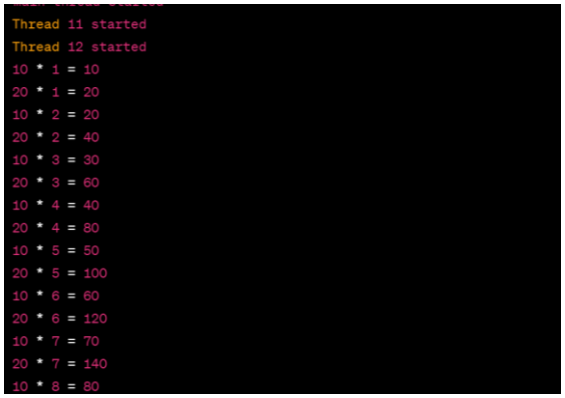
43. Write a program to find the number of special characters in the given statement
Sample Input:
Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.
Sample Output:
Number of special Characters: 5

```
import java.util.Scanner;
public class CountSpecialCharacters {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a statement: ");
      String statement = scanner.nextLine();
      int specialCharCount = countSpecialCharacters(statement);
      System.out.println("Number of special Characters: " + specialCharCount);
   }
   public static int countSpecialCharacters(String str) {
      int specialCount = 0;
      for (int i = 0; i < str.length(); i++) {
         char ch = str.charAt(i);
         if (!Character.isLetterOrDigit(ch) && !Character.isWhitespace(ch)) {
            specialCount++;
         }
      }

      return specialCount;
   }
}
```

```
Enter a statement: Modi Birthday @ September 17, #&$% is the wishes code fo
Number of special Characters: 5
```

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

```
class MultiplicationTableThread extends Thread {
    private int number;

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

    @Override
    public void run() {
        System.out.println("Thread " + Thread.currentThread().getId() + " started");
        for (int i = 1; i <= 10; i++) {
            System.out.println(number + " * " + i + " = " + (number * i));
        }
        System.out.println("Thread " + Thread.currentThread().getId() + " finished");
    }
}

public class ThreadLifecycleExample {
    public static void main(String[] args) {
        MultiplicationTableThread table5Thread = new MultiplicationTableThread(5);
        MultiplicationTableThread table10Thread = new MultiplicationTableThread(10);

        System.out.println("Main thread started");

        table5Thread.start();  // Thread enters the "Runnable" state and starts executing run()
        table10Thread.start(); // Thread enters the "Runnable" state and starts executing run()

        try {
            table5Thread.join();  // Wait for table5Thread to complete
            table10Thread.join(); // Wait for table10Thread to complete
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("Main thread finished");
    }
}
```



45. 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 PatternPrinter {

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

      System.out.print("Enter the number to be printed: ");
      int num = sc.nextInt();

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

      printPattern(num, maxTimes);

      sc.close();
   }

   public static void printPattern(int num, int maxTimes) {
      for (int i = 1; i <= maxTimes; i++) {
         for (int j = 1; j <= i; j++) {
            System.out.print(num);
         }
         System.out.println();
      }
```

```
        for (int i = maxTimes - 1; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print(num);
            }
            System.out.println();
        }
    }
}
```

```
Enter the number to be printed: 1
Max Number of times printed: 3
```

```
1
11
111
11
1
```

46. 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

```
import java.util.Scanner;

public class LeapYearCheck {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter Date (MM/DD/YYYY): ");

        String date = scanner.nextLine();

        int year = extractYear(date);

        if (isLeapYear(year)) {

            System.out.println("Given year is a Leap Year");

        } else {

            System.out.println("Given year is a Non Leap Year");

        }

    }
```

```
    public static int extractYear(String date) {
        String[] parts = date.split("/");
        return Integer.parseInt(parts[2]);
    }
    public static boolean isLeapYear(int year) {
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            return true;
        } else {
            return false;
        }
    }
}
```

```
Enter Date: 04/11/1947
```

```
Given year is a Non Leap Year
```