INSTITUTE OF SCIENCE & TECHNOLOGY
FOR ADVANCED STUDIES & RESEARCH

ISTAR-CONSTITUENT COLLEGE OF CVM
UNIVERSITY

DEPARTMENT MASTER OF COMPUTER APPLICATION

LAB MANUAL

FOR

PAPER CODE-PAPER TITLE

SEMESTER I

ACADEMIC YEAR

2024-25

Prepared By:	Reviewed By:
Student Name	Dr. Niky K. Jain
Enrollement No.	Asst. Professor
MCA	Computer Science Department
ISTAR	ISTAR

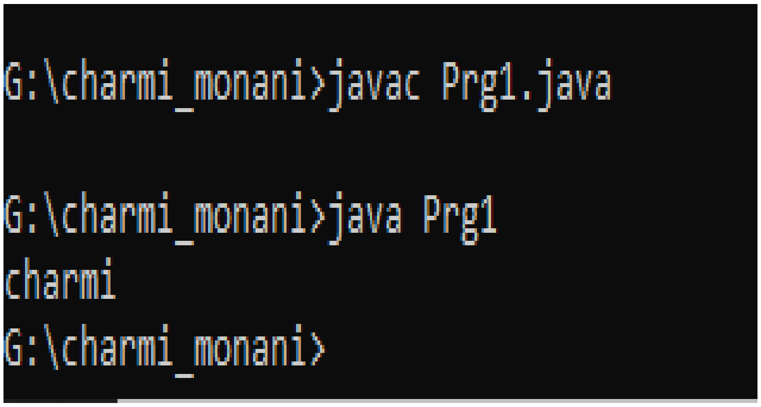
<table>
<tr><td>SR.
NO</td><td>PROGRAM</td></tr>
<tr><td><b>1</b></td><td>

Write a program to print Your Name (Without using semicolon).

Soln:

```
class Study
{
        public static void main(String args[])
        {
        if(System.out.printf("charmi")==null)
        {
        }
        }
}
```

Output:



```
G:\charmi_monani>javac Prg1.java

G:\charmi_monani>java Prg1
charmi
G:\charmi_monani>
```

</td></tr>
<tr><td><b>2</b></td><td>

Write a program to find no is odd or even.

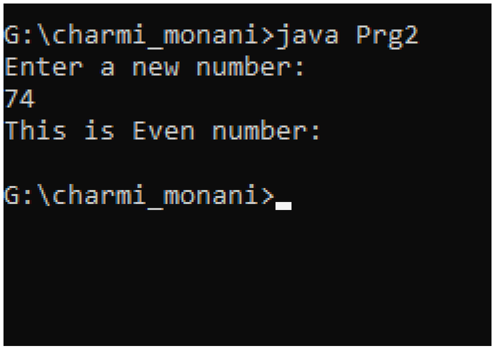
Soln:

```
import java.util.Scanner;
class OddEven
{
        public static void main(String args[])
        {
                int n;
                System.out.println("Enter a new number:");
                Scanner r=new Scanner(System.in);
                n = r.nextInt();
                if(n%2==0)
                {
                        System.out.println("This is Even number:");
                }
                else
                {
```

</td></tr>
</table>

```
                                        System.out.println("This is Odd number:");
                        }
            }
}
```

Output:



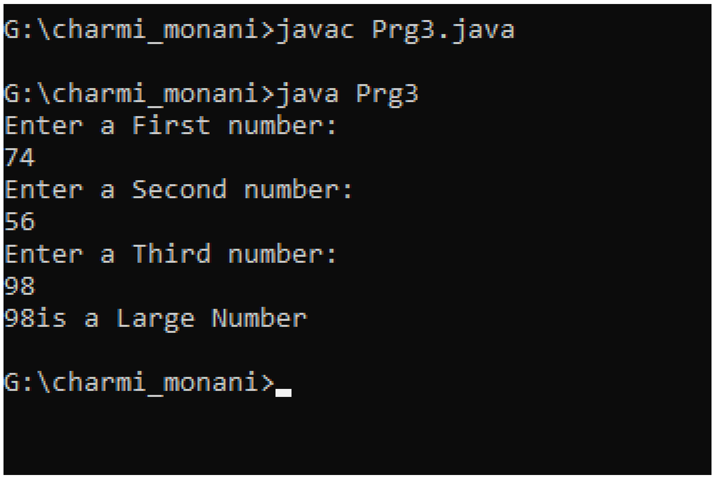
```
G:\charmi_monani>java Prg2
Enter a new number:
74
This is Even number:

G:\charmi_monani>_
```

3 **Write a program to find greater among three number.**

Soln:

```
import java.util.Scanner;
public class LargestOfNumber
{
            public static void main(String args[])
            {
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter a First number:");
            int a =sc.nextInt();
            System.out.println("Enter a Second number:");
            int b =sc.nextInt();
            System.out.println("Enter a Third number:");
            int c =sc.nextInt();
            if(a>b && a>c)
            {
                        System.out.println(a+"is a Large Number");
            }
            else if(b>a && b>c)
            {
                        System.out.println(b+"is a Large Number");
            }
            else
            {
                        System.out.println(c+"is a Large Number");

            }

            }
}
```

Output:



```
G:\charmi_monani>javac Prg3.java

G:\charmi_monani>java Prg3
Enter a First number:
74
Enter a Second number:
56
Enter a Third number:
98
98is a Large Number

G:\charmi_monani>
```

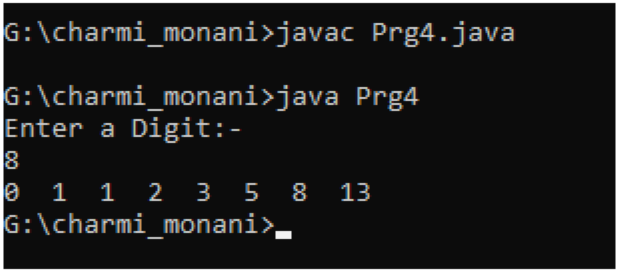
4

Write a program to generate Fibonacci series.

Soln:

```
import java.util.Scanner;
class Fibonacci
{
        public static void main(String args[])
        {
                Scanner sc=new Scanner(System.in);
                int t1=0,t2=1;
                System.out.println("Enter a Digit:-");
                int n=sc.nextInt();
                for(int i=1;i<=n;i++)
                {
                        System.out.print(t1+"  ");
                        int sum=t1+t2;
                        t1=t2;
                        t2=sum;

                }

        }
}
```

Output:



```
G:\charmi_monani>javac Prg4.java

G:\charmi_monani>java Prg4
Enter a Digit:-
8
0  1  1  2  3  5  8  13
G:\charmi_monani>
```

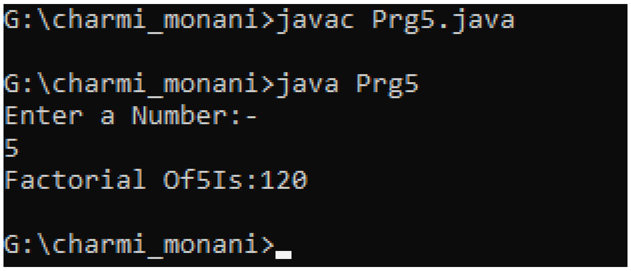
5

Write a program find a factorial of a Number using recursion.

Soln:

```
import java.util.Scanner;
class Factorial
{
        static int factorial(int n)
        {
        if(n==0 || n==1)
                return 1;
                return n*factorial(n-1);
        }

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

               System.out.println("Enter a Number:-");
               int number = p.nextInt();

               int result = factorial(number);

               System.out.println("Factorial Of"+number+"Is:"+result);


       }
}
```

Output:



```
G:\charmi_monani>javac Prg5.java

G:\charmi_monani>java Prg5
Enter a Number:-
5
Factorial Of5Is:120

G:\charmi_monani>
```

6

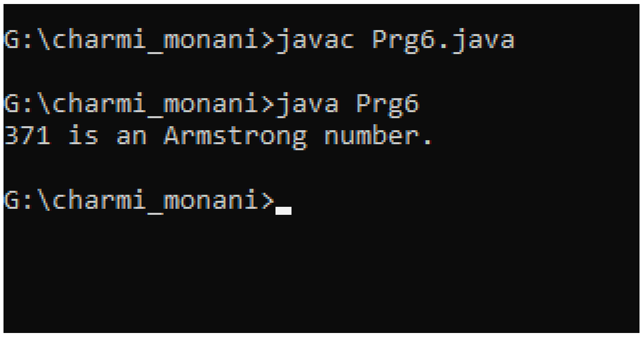
Write a program to find the number is Armstrong or not.

Soln:

```
public class Prg6
```

```
{

  public static void main(String[] args) {

    int number = 371, originalNumber, remainder, result = 0;

    originalNumber = number;

    while (originalNumber != 0)
    {
      remainder = originalNumber % 10;
      result += Math.pow(remainder, 3);
      originalNumber /= 10;
    }

    if(result == number)
      System.out.println(number + " is an Armstrong number.");
    else
      System.out.println(number + " is not an Armstrong number.");
  }
}
```

Output:



```
G:\charmi_monani>javac Prg6.java

G:\charmi_monani>java Prg6
371 is an Armstrong number.

G:\charmi_monani>_
```

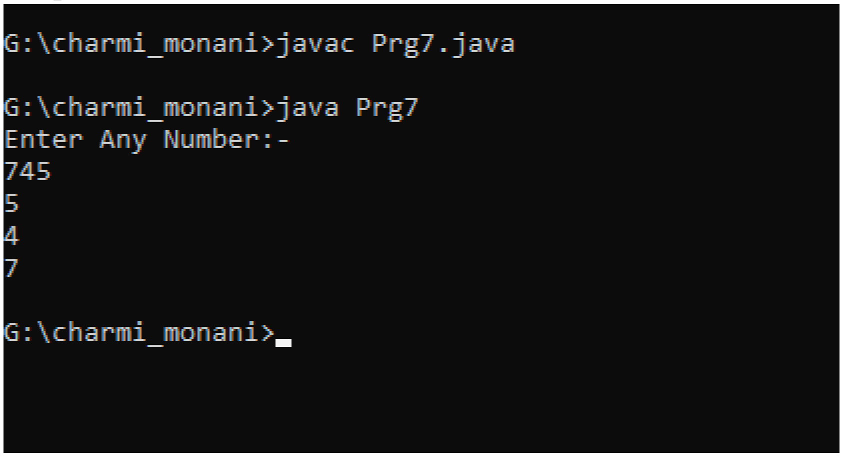
7 **Write a program to find the reverse of n number.**

Soln:

```
import java.util.Scanner;
class Prg7
{
        public static void main(String args[])
        {
                int n,r;
                System.out.println("Enter Any Number:-");
                Scanner ref=new Scanner(System.in);
                n=ref.nextInt();
                while(n>0)
                {
                        r=n%10;
                        System.out.println(r);
```

```
                                n=n/10;
                      }
            }

}
```

Output:



```
G:\charmi_monani>javac Prg7.java

G:\charmi_monani>java Prg7
Enter Any Number:-
745
5
4
7

G:\charmi_monani>_
```

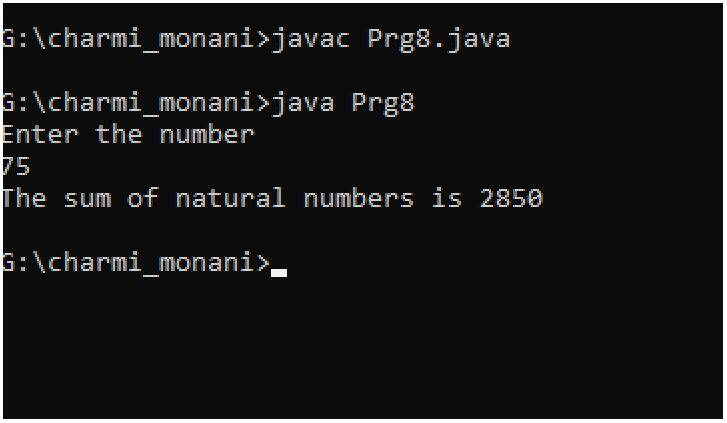
8

Write a program to find addition of n number.

Soln:

```
import java.util.Scanner;
public class Prg8
{


    public static void main(String []args)
    {
      Scanner sc=new Scanner(System.in);
      int num;
      System.out.println("Enter the number");
      num=sc.nextInt();

      int sum=0;
      for(int i=1; i<=num;i++)
      {
         sum=sum+i;
      }
      System.out.println("The sum of natural numbers is "+sum);
    }
}
```

Output:



```
G:\charmi_monani>javac Prg8.java

G:\charmi_monani>java Prg8
Enter the number
75
The sum of natural numbers is 2850

G:\charmi_monani>_
```

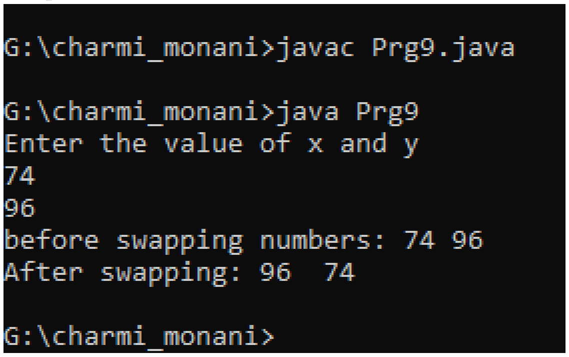
9

Write a program swapping of two numbers without using third variable.

Soln:

```
import java.util.Scanner;
class Prg9
{
   public static void main(String a[])
   {
      System.out.println("Enter the value of x and y");
      Scanner sc = new Scanner(System.in);

      int x = sc.nextInt();
      int y = sc.nextInt();
      System.out.println("before swapping numbers: "+x +" "+ y);
      x = x + y;
      y = x - y;
      x = x - y;
      System.out.println("After swapping: "+x +"  " + y);
   }
}
```

Output:



```
G:\charmi_monani>javac Prg9.java

G:\charmi_monani>java Prg9
Enter the value of x and y
74
96
before swapping numbers: 74 96
After swapping: 96  74

G:\charmi_monani>
```

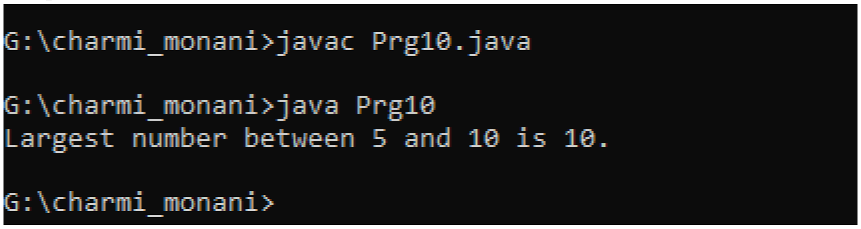
10

Find out greater number between three numbers of using condition operator.

Soln:

```
class Prg10
{
public static void main(String args[])
{

        int n1 = 5, n2 = 10, max;


        max = (n1 > n2) ? n1 : n2;


        System.out.println("Largest number between " + n1 +
                                " and " + n2 + " is " + max + ". " );
}
}
```

Output:



```
G:\charmi_monani>javac Prg10.java

G:\charmi_monani>java Prg10
Largest number between 5 and 10 is 10.

G:\charmi_monani>
```

11 **Write a java program to implement all the Java Operators.**

Soln:

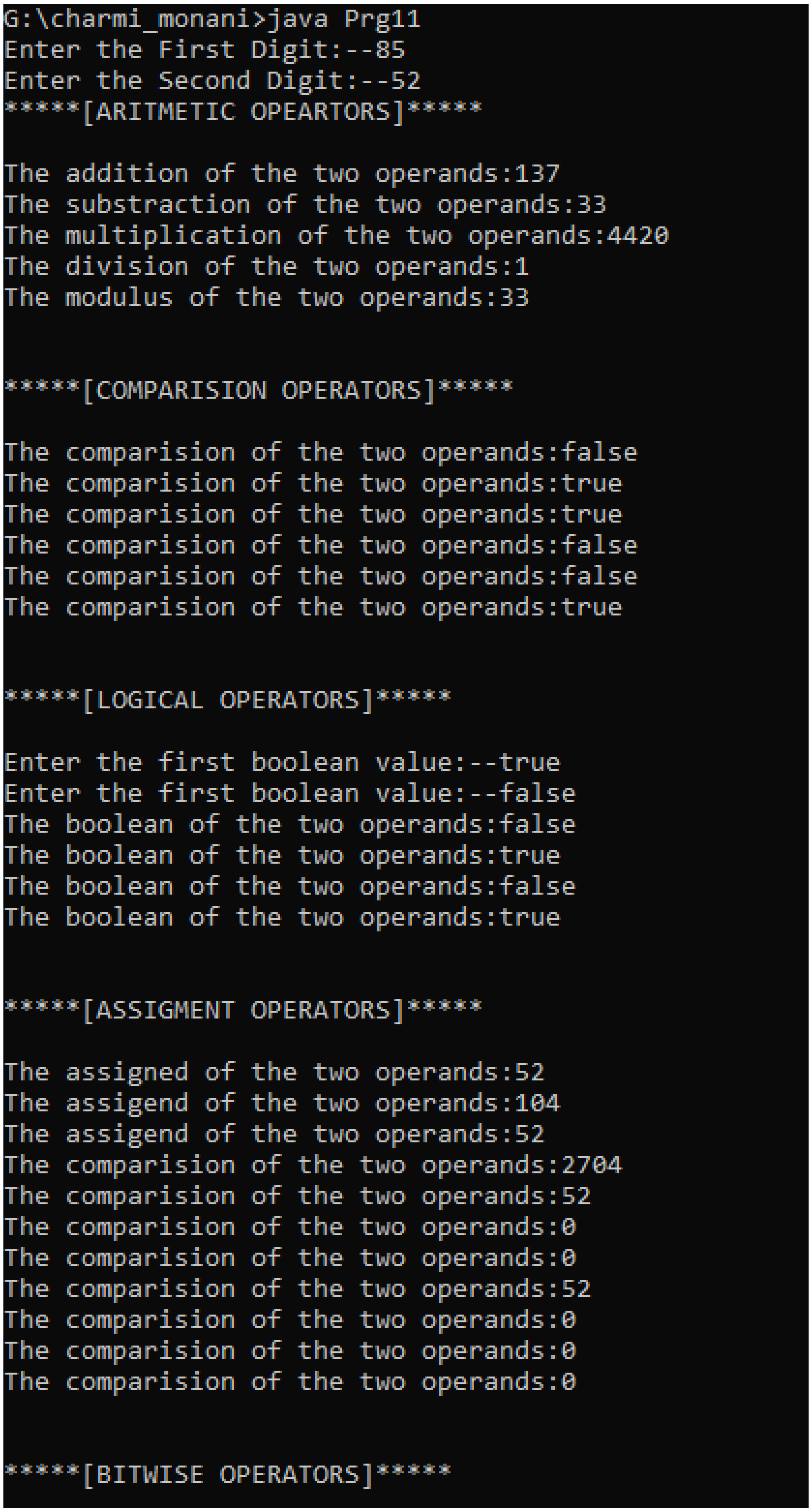
```
import java.util.Scanner;
class Prg11
{
        public static void main(String args[])
        {
                Prg11 star=new Prg11();
                Scanner s = new Scanner(System.in);
                {
                        System.out.print("Enter the First Digit:--");
                        int p = s.nextInt();
                        System.out.print("Enter the Second Digit:--");
                        int q = s.nextInt();
                        System.out.println("*****[ARITMETIC OPEARTORS]*****\n");
                        System.out.println("The addition of the two operands:"+(p+q));
                        System.out.println("The substraction of the two
operands:"+(p-q));
                        System.out.println("The multiplication of the two
operands:"+(p*q));
                        System.out.println("The division of the two operands:"+(p/q));
```

```
                                System.out.println("The modulus of the two
operands:"+(p%q));
                                System.out.println("\n");



                                System.out.println("*****[COMPARISION
OPERATORS]*****\n");
                                System.out.println("The comparision of the two
operands:"+(p==q));
                                System.out.println("The comparision of the two
operands:"+(p!=q));
                                System.out.println("The comparision of the two
operands:"+(p>q));
                                System.out.println("The comparision of the two
operands:"+(p<q));
                                System.out.println("The comparision of the two
operands:"+(p<=q));
                                System.out.println("The comparision of the two
operands:"+(p>=q));
                                System.out.println("\n");

                                System.out.println("*****[LOGICAL OPERATORS]*****\n");
                                System.out.print("Enter the first boolean value:--");
                                boolean a = s.nextBoolean();
                                System.out.print("Enter the first boolean value:--");
                                boolean b = s.nextBoolean();
                                System.out.println("The boolean of the two
operands:"+(a&&b)); //AND
                                System.out.println("The boolean of the two operands:"+(a||b));
//OR
                                System.out.println("The boolean of the two operands:"+(!a));
//NOT
                                System.out.println("The boolean of the two operands:"+(!b));
//NOT
                                System.out.println("\n");

                                System.out.println("*****[ASSIGMENT OPERATORS]*****\n");
                                System.out.println("The assigned of the two operands:"+(p=q));
                                System.out.println("The assigend of the two
operands:"+(p+=q)); //p = p + q
                                System.out.println("The assigend of the two
operands:"+(p-=q)); //p = p - q
                                System.out.println("The comparision of the two
operands:"+(p*=q)); //p = p * q
                                System.out.println("The comparision of the two
operands:"+(p/=q));
                                System.out.println("The comparision of the two
operands:"+(p%=q));
                        System.out.println("The comparision of the two operands:"+(p&=q));
```

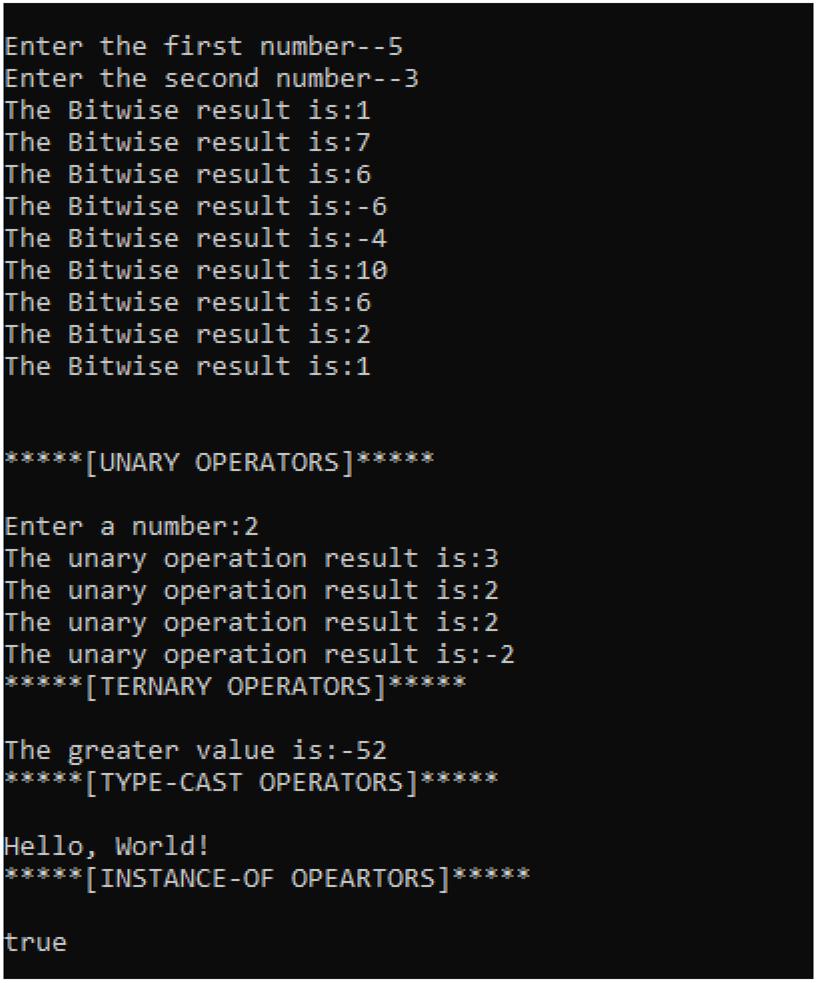
```
                                    System.out.println("The comparision of the two 
operands:"+(p|=q));
                                    System.out.println("The comparision of the two 
operands:"+(p^=q));
                                    System.out.println("The comparision of the two 
operands:"+(p>>=q));
                                    System.out.println("The comparision of the two 
operands:"+(p<<=q));
                                    System.out.println("\n");

                                    System.out.println("*****[BITWISE OPERATORS]*****\n");
                                    System.out.print("Enter the first number--");
                                    int c = s.nextInt();
                                    System.out.print("Enter the second number--");
                                    int d = s.nextInt();
                                    System.out.println("The Bitwise result is:"+(c&d));
                                    System.out.println("The Bitwise result is:"+(c|d));
                                    System.out.println("The Bitwise result is:"+(c^d));
                                    System.out.println("The Bitwise result is:"+(~c));
                                    System.out.println("The Bitwise result is:"+(~d));
                                    System.out.println("The Bitwise result is:"+(c<<1));
                                    System.out.println("The Bitwise result is:"+(d<<1));
                                    System.out.println("The Bitwise result is:"+(c>>1));
          System.out.println("The Bitwise result is:"+(d>>1));
                                    System.out.println("\n");

                                    System.out.println("*****[UNARY OPERATORS]*****\n");
                                    System.out.print("Enter a number:");
                                    int h = s.nextInt();
                                    System.out.println("The unary operation result is:"+(++h));
                                    System.out.println("The unary operation result is:"+(--h));
                                    System.out.println("The unary operation result is:"+(+h));
                                    System.out.println("The unary operation result is:"+(-h));


                                    System.out.println("*****[TERNARY OPERATORS]*****\n");
                                    int ter = (p>q)?p:q;
          System.out.println("The greater value is:-" +ter);


                                    System.out.println("*****[TYPE-CAST OPERATORS]*****\n");
                                    Object obj = "Hello, World!";
                                    String str = (String) obj;
                                    System.out.println(str);

                        System.out.println("*****[INSTANCE-OF OPEARTORS]*****\n");
                                    System.out.println(star instanceof Prg11);
                        }
                        s.close();
            }
```

}

Output:



```
G:\charmi_monani>java Prg11
Enter the First Digit:--85
Enter the Second Digit:--52
*****[ARITMETIC OPEARTORS]*****

The addition of the two operands:137
The substraction of the two operands:33
The multiplication of the two operands:4420
The division of the two operands:1
The modulus of the two operands:33


*****[COMPARISION OPERATORS]*****

The comparision of the two operands:false
The comparision of the two operands:true
The comparision of the two operands:true
The comparision of the two operands:false
The comparision of the two operands:false
The comparision of the two operands:true


*****[LOGICAL OPERATORS]*****

Enter the first boolean value:--true
Enter the first boolean value:--false
The boolean of the two operands:false
The boolean of the two operands:true
The boolean of the two operands:false
The boolean of the two operands:true


*****[ASSIGMENT OPERATORS]*****

The assigned of the two operands:52
The assigend of the two operands:104
The assigend of the two operands:52
The comparision of the two operands:2704
The comparision of the two operands:52
The comparision of the two operands:0
The comparision of the two operands:0
The comparision of the two operands:52
The comparision of the two operands:0
The comparision of the two operands:0
The comparision of the two operands:0


*****[BITWISE OPERATORS]*****
```



```
Enter the first number--5
Enter the second number--3
The Bitwise result is:1
The Bitwise result is:7
The Bitwise result is:6
The Bitwise result is:-6
The Bitwise result is:-4
The Bitwise result is:10
The Bitwise result is:6
The Bitwise result is:2
The Bitwise result is:1


*****[UNARY OPERATORS]*****

Enter a number:2
The unary operation result is:3
The unary operation result is:2
The unary operation result is:2
The unary operation result is:-2
*****[TERNARY OPERATORS]*****

The greater value is:-52
*****[TYPE-CAST OPERATORS]*****

Hello, World!
*****[INSTANCE-OF OPEARTORS]*****

true
```

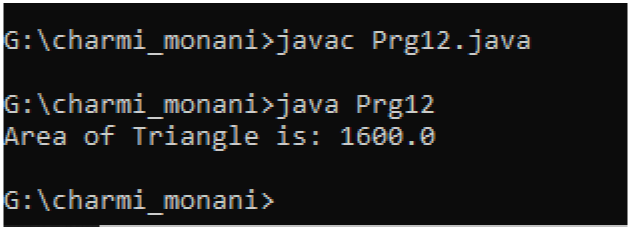
12

Find out area of triangle.

Soln:

```
class Prg12
{
        public static void main(String args[])
        {

                float b=40,h=80,area;

                 area = ( b*h) / 2 ;
     System.out.println("Area of Triangle is: "+area);
        }
}
```

Output:



```
G:\charmi_monani>javac Prg12.java

G:\charmi_monani>java Prg12
Area of Triangle is: 1600.0

G:\charmi_monani>
```

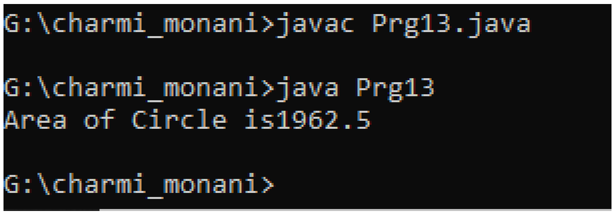
13

Write a program find calculate area of circle using its radius.

Soln:

```
class Prg13
{
        public static void main(String args[])
        {
                int radius;
                double pi=3.14,area;
                radius=25;

                area=pi*radius*radius;
                System.out.println("Area of Circle is"+area);
        }
}
```

Output:



```
G:\charmi_monani>javac Prg13.java

G:\charmi_monani>java Prg13
Area of Circle is1962.5

G:\charmi_monani>
```

14

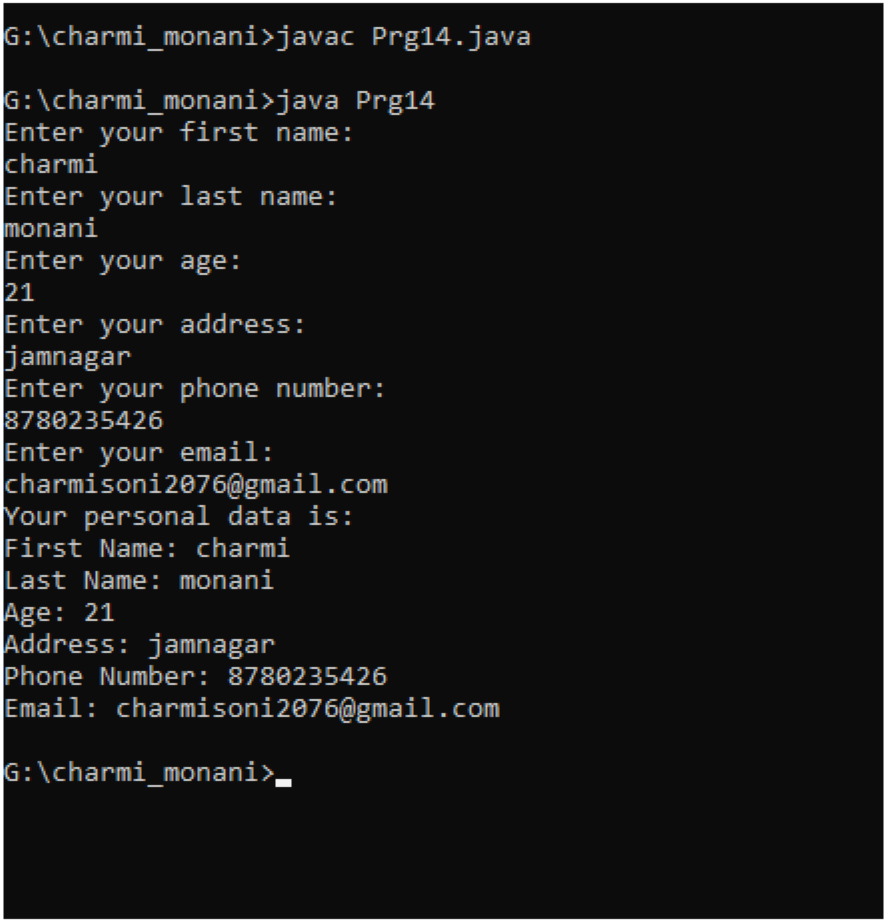
Write a java program to take persnol data from user.

Soln:

```
import java.util.Scanner;
class Prg14
        {
        public static void main(String args[])
                {

    Scanner scanner = new Scanner(System.in);

    System.out.println("Enter your first name: ");
    String firstName = scanner.nextLine();

    System.out.println("Enter your last name: ");
    String lastName = scanner.nextLine();
```

```
        System.out.println("Enter your age: ");
        int age = scanner.nextInt();

        System.out.println("Enter your address: ");
        String address = scanner.next();

        System.out.println("Enter your phone number: ");
        String phoneNumber = scanner.next();

        System.out.println("Enter your email: ");
        String email = scanner.next();

        System.out.println("Your personal data is: ");
        System.out.println("First Name: " + firstName);
        System.out.println("Last Name: " + lastName);
        System.out.println("Age: " + age);
        System.out.println("Address: " + address);
        System.out.println("Phone Number: " + phoneNumber);
        System.out.println("Email: " + email);

}

}
```

Output:



```
G:\charmi_monani>javac Prg14.java

G:\charmi_monani>java Prg14
Enter your first name:
charmi
Enter your last name:
monani
Enter your age:
21
Enter your address:
jamnagar
Enter your phone number:
8780235426
Enter your email:
charmisoni2076@gmail.com
Your personal data is:
First Name: charmi
Last Name: monani
Age: 21
Address: jamnagar
Phone Number: 8780235426
Email: charmisoni2076@gmail.com

G:\charmi_monani>_
```

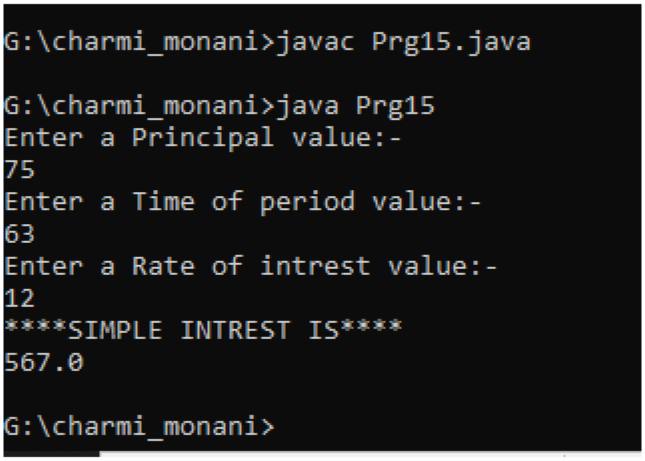
15

Write a program to find simple Intrest.

Soln:

```
import java.util.Scanner;
class Prg15
{
        public static void main(String args[])
        {
                int p,r,t;
                Scanner s1=new Scanner(System.in);
                System.out.println("Enter a Principal value:-");
                p=s1.nextInt();
                System.out.println("Enter a Time of period value:-");
                r=s1.nextInt();
                System.out.println("Enter a Rate of intrest value:-");
                t=s1.nextInt();
                float simpleintrest;
                simpleintrest=(p*r*t)/100;
                System.out.println("****SIMPLE INTREST IS**** \n"+simpleintrest);

        }
}
```

Output:



```
G:\charmi_monani>javac Prg15.java

G:\charmi_monani>java Prg15
Enter a Principal value:-
75
Enter a Time of period value:-
63
Enter a Rate of intrest value:-
12
****SIMPLE INTREST IS****
567.0

G:\charmi_monani>
```

16A

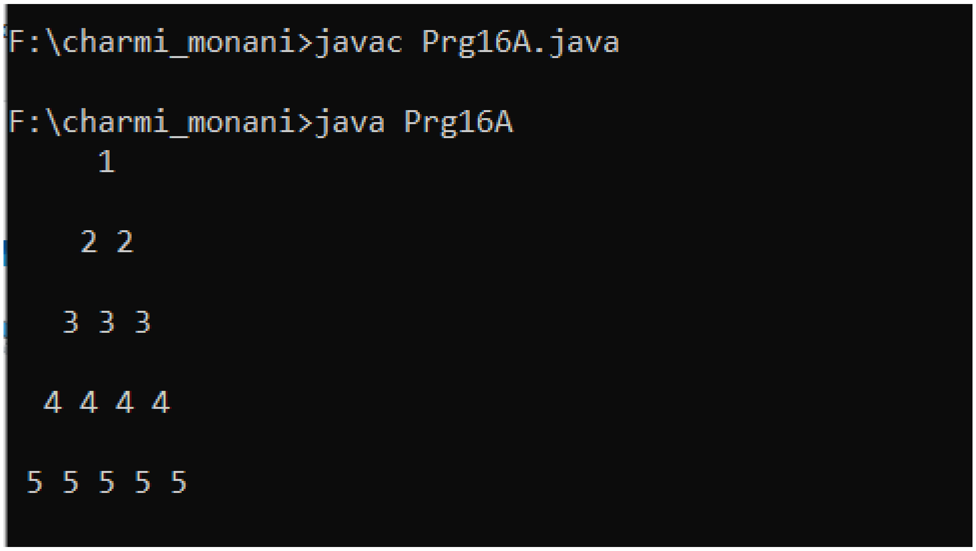
Write a Program to generate series.

Soln:

```
class Prg16A

{

        public static void main(String[] args)

        {

                int i,j;

                int n=5;

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

                {

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

                                {

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

```
                        }
                        for(int m=1;m<=i;m++)
                        {
                              System.out.print(i+" ");
                        }
                        System.out.println("\n");
            }
      }
}
```

Output:



```
F:\charmi_monani>javac Prg16A.java

F:\charmi_monani>java Prg16A
     1

    2 2

   3 3 3

  4 4 4 4

 5 5 5 5 5
```

16B

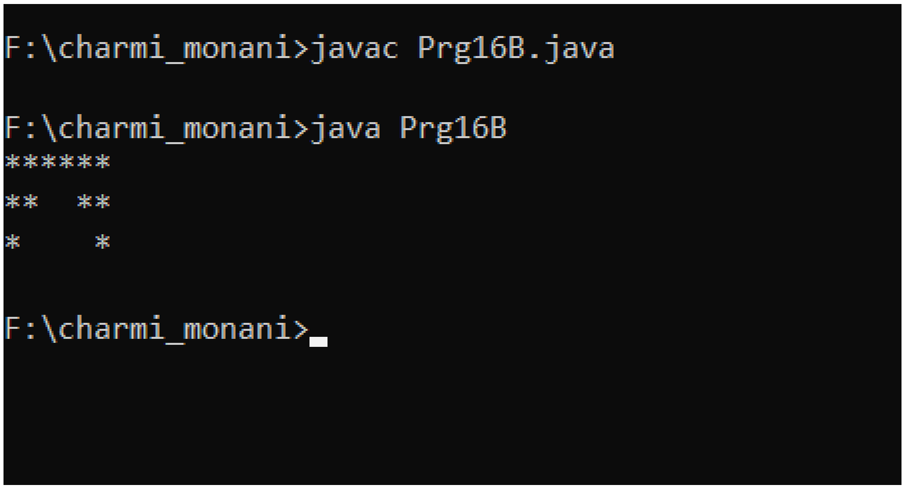
Write a Program to generate series.

Soln:

```
import java.util.*;
 class Prg16B
{
   public static void main(String args[])
   {
      Scanner s1=new Scanner(System.in);
                   int num=6;
                   int number=num/2;
      for(int i=1; i<=number; i++)
      {
         for(int j=1; j<=number-i+1; j++)
         {
            System.out.print("*");
         }

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

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


    }
}
```

Output:



```
F:\charmi_monani>javac Prg16B.java

F:\charmi_monani>java Prg16B
******
**  **
*    *

F:\charmi_monani>_
```

16C

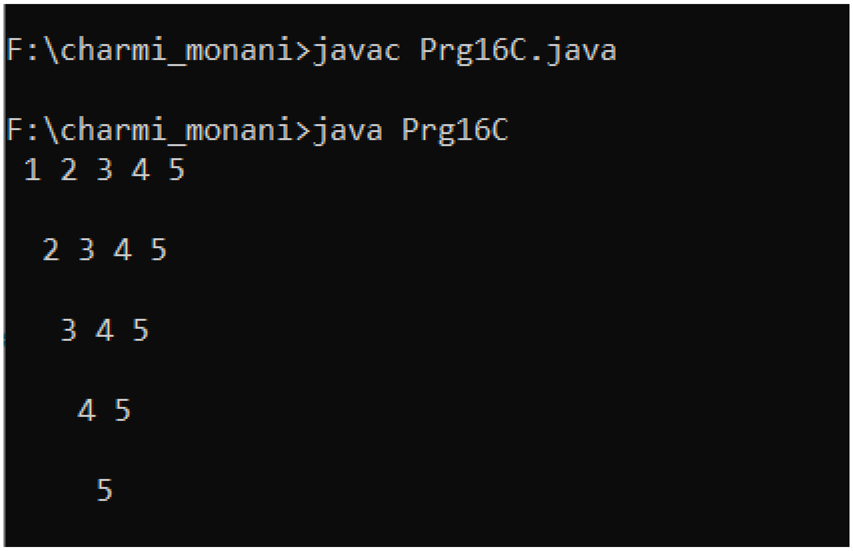
Write a Program to generate series.

Soln:

```
class Prg16C
{
        public static void main(String[] args)
        {
                int i,j;
                int n=5;
                for (i=1; i<=n; i++)
                {
```

```
                for(j=1;j<=i;j++)
                {
                    System.out.print(" ");
                }
                for(int m=i;m<=n;m++)
                {
                    System.out.print(m+" ");
                }
                System.out.println("\n");
        }
    }
}
```

OUTPUT:



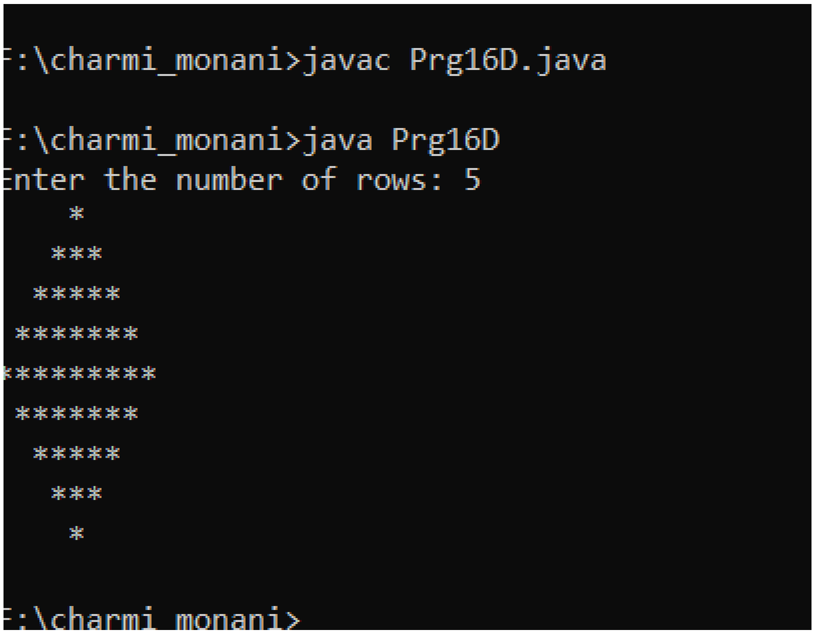
```
F:\charmi_monani>javac Prg16C.java

F:\charmi_monani>java Prg16C
 1 2 3 4 5

  2 3 4 5

   3 4 5

    4 5

     5
```

16D	**Write a Program to generate series.**

Soln:

```
import java.util.Scanner;


public class Prg16D

{

    public static void main(String[] args)

              {

        Scanner scanner = new Scanner(System.in);

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

        int rows = scanner.nextInt();




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

                            {

            for (int j = rows; j > i; j--)

                                                       {

                System.out.print(" ");

            }

            for (int k = 1; k <= (2 * i - 1); k++)

                                                                     {

                System.out.print("*");
            }
            System.out.println();
        }



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

    }
}
```

Output:



```
F:\charmi_monani>javac Prg16D.java

F:\charmi_monani>java Prg16D
Enter the number of rows: 5
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *

F:\charmi_monani>
```

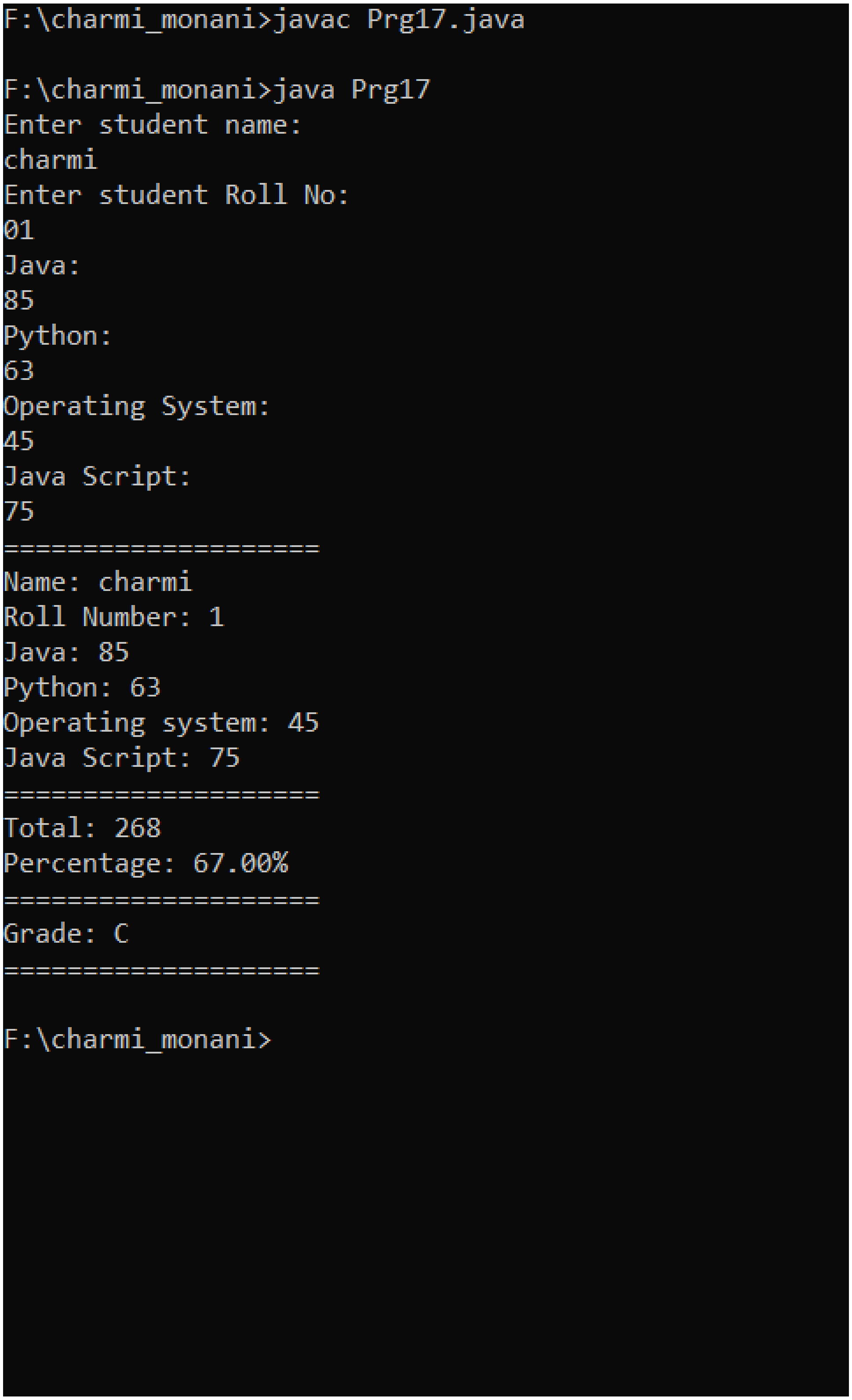
17 **WAP to Print the mark sheet of a student .**

Soln:

```
import java.util.Scanner;
 class Prg17
{
   public static void main(String[] args)
	{
      Scanner s1 = new Scanner(System.in);
		System.out.println("Enter student name: ");
		String name=s1.next();
		System.out.println("Enter student Roll No: ");
		int rollno=s1.nextInt();
		System.out.println("Java: ");
		int java=s1.nextInt();
		System.out.println("Python: ");
		int python=s1.nextInt();
		System.out.println("Operating System: ");
		int os=s1.nextInt();
		System.out.println("Java Script: ");
		int js=s1.nextInt();

		int total=java+python+os+js;
		double per=((double)total/400*100);

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

```
                System.out.println("Name: "+name);
                System.out.println("Roll Number: "+rollno);
                System.out.println("Java: "+java);
                System.out.println("Python: "+python);
                System.out.println("Operating system: "+os);
                System.out.println("Java Script: "+js);
                System.out.println("====================");
    System.out.println("Total: "+total);
                System.out.println("Percentage: "+String.format("%.2f",per)+"%");
                System.out.println("====================");
    // Determine grade
    if (per >= 90)
                {
      System.out.println("Grade: A+");
    }
                else if (per >= 80)
                {
      System.out.println("Grade: A");
    }
                else if (per >= 70)
                {
      System.out.println("Grade: B");
    }
                else if (per >= 60)
                {
      System.out.println("Grade: C");
    }
                else
                {
      System.out.println("fail");
    }
                System.out.println("====================");


  }
}
```

Output:



```
F:\charmi_monani>javac Prg17.java

F:\charmi_monani>java Prg17
Enter student name:
charmi
Enter student Roll No:
01
Java:
85
Python:
63
Operating System:
45
Java Script:
75
====================
Name: charmi
Roll Number: 1
Java: 85
Python: 63
Operating system: 45
Java Script: 75
====================
Total: 268
Percentage: 67.00%
====================
Grade: C
====================

F:\charmi_monani>
```

18	**To find out the volume of the two different boxes having following values.**

	Width	**Height**	**Depth**
Box1:	**110**	**20**	**15**
Box2:	**23**	**6**	**9**

Value :

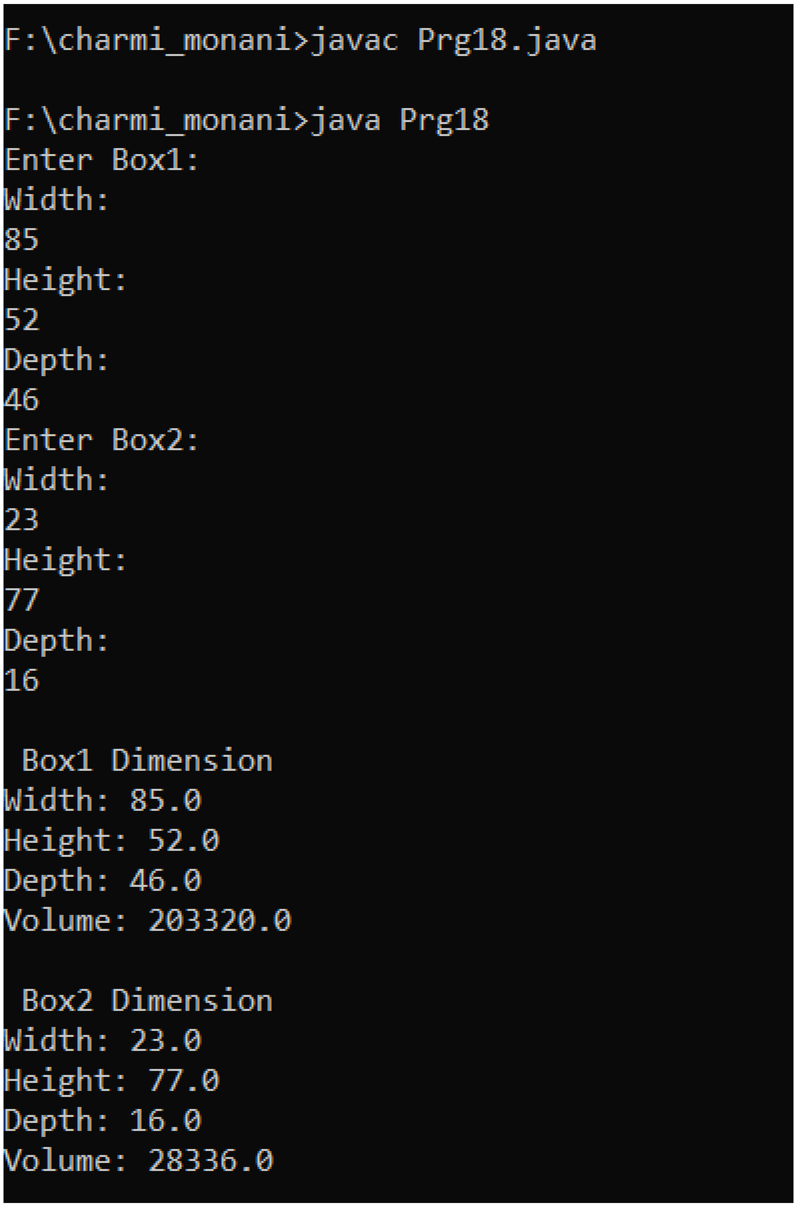
Soln:

```
import java.util.Scanner;
 class Prg18
{
   public static void main(String[] args)
          {
      Scanner s1 = new Scanner(System.in);
                    System.out.println("Enter Box1: ");
                    System.out.println("Width: ");
                    double width1=s1.nextDouble();
                    System.out.println("Height: ");
                    double height1=s1.nextDouble();
                    System.out.println("Depth: ");
                    double depth1=s1.nextDouble();
                    System.out.println("Enter Box2: ");
                    System.out.println("Width: ");
                    double width2=s1.nextDouble();
```

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

        double height2=s1.nextDouble();

        System.out.println("Depth: ");

        double depth2=s1.nextDouble();


        double volume1=calculateVolume(width1,height1,depth1);

        double volume2=calculateVolume(width2,height2,depth2);


        System.out.println("\n Box1 Dimension");

        System.out.println("Width: "+width1);

        System.out.println("Height: "+height1);

        System.out.println("Depth: "+depth1);

        System.out.println("Volume: "+volume1);

        System.out.println("\n Box2 Dimension");

        System.out.println("Width: "+width2);

        System.out.println("Height: "+height2);

        System.out.println("Depth: "+depth2);

        System.out.println("Volume: "+volume2);

    }

    public static double calculateVolume(double width,double height,double
depth)

    {

        return width*height*depth;
```

```
      }

}
```

OUTPUT:



```
F:\charmi_monani>javac Prg18.java

F:\charmi_monani>java Prg18
Enter Box1:
Width:
85
Height:
52
Depth:
46
Enter Box2:
Width:
23
Height:
77
Depth:
16

 Box1 Dimension
Width: 85.0
Height: 52.0
Depth: 46.0
Volume: 203320.0

 Box2 Dimension
Width: 23.0
Height: 77.0
Depth: 16.0
Volume: 28336.0
```

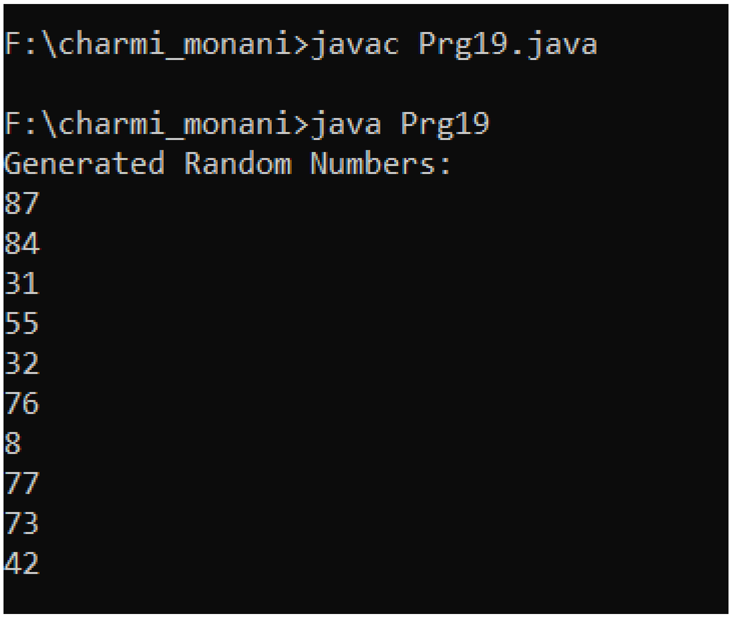
19

Create a java program to generate ten random number store and display them.

Soln:

```
import java.util.Random;

class Prg19

{

        public static void main(String args[])

        {

                Random r1=new Random();

                int[]randomNumber=new int[10];

                for(int i=0;i<10;i++)

                {

                        randomNumber[i]=r1.nextInt(100);

                }

                System.out.println("Generated Random Numbers: ");

                for(int number:randomNumber)

                {

                        System.out.println(number+" ");

                }

        }

}
```

Output:



```
F:\charmi_monani>javac Prg19.java

F:\charmi_monani>java Prg19
Generated Random Numbers:
87
84
31
55
32
76
8
77
73
42
```

20

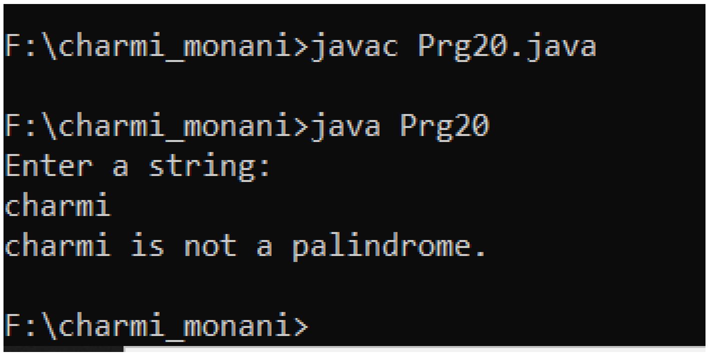
WAP to take string as a command line argument and check whether the string is palindrome or not.

Soln:

```
import java.util.Scanner;


public class Prg20

{

   public static void main(String[] args)

          {

      Scanner scanner = new Scanner(System.in);

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

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

```
    input = input.toLowerCase();



    String reversed = new StringBuilder(input).reverse().toString();



    if (input.equals(reversed))
              {
      System.out.println(input + " is a palindrome.");
    }
              else
              {
      System.out.println(input + " is not a palindrome.");
    }

  }
}
```

Output:



```
F:\charmi_monani>javac Prg20.java

F:\charmi_monani>java Prg20
Enter a string:
charmi
charmi is not a palindrome.

F:\charmi_monani>
```

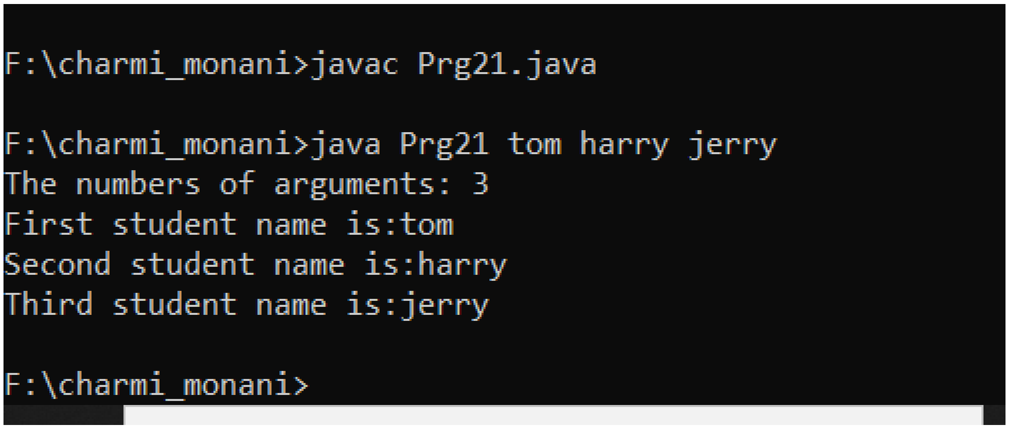
p21

Write a java application which takes several command line arguments , which are suppose to be name of student and prints.

Soln:

```
import java.util.*;

public class Prg21

{

        public static void main(String args[])

        {

                System.out.println("The numbers of arguments: "+args.length);

                String a[]={"First","Second","Third"};

                for(int i=0;i<3;i++)

                {

                                System.out.println(a[i]+" "+ "student name is:"+args[i]);

                }

        }
}
```

Output:



```
F:\charmi_monani>javac Prg21.java

F:\charmi_monani>java Prg21 tom harry jerry
The numbers of arguments: 3
First student name is:tom
Second student name is:harry
Third student name is:jerry

F:\charmi_monani>
```

p22

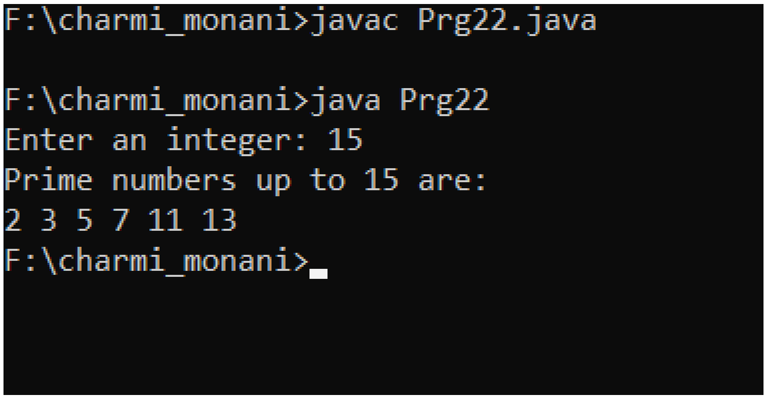
Write a java program that prompts the user for an integer and then prints our all the prime numbers up to that integer

Soln:

```
import java.util.Scanner;

public class Prg22

{

  public static void main(String[] args)

       {

    Scanner s1 = new Scanner(System.in);

    System.out.print("Enter an integer: ");

    int a = s1.nextInt();

    System.out.println("Prime numbers up to " + a + " are:");
    for (int i = 2; i <= a; i++)
             {
      if (isPrime(i))
                           {
        System.out.print(i + " ");
      }
    }
  }

  public static boolean isPrime(int n)
       {
    if (n <= 1)
             {
      return false;
    }
    for (int i = 2; i <= Math.sqrt(n); i++)
             {
      if (n % i == 0)
                           {
        return false;
      }
    }
```

```
        return true;

    }

}
```

Output:



```
F:\charmi_monani>javac Prg22.java

F:\charmi_monani>java Prg22
Enter an integer: 15
Prime numbers up to 15 are:
2 3 5 7 11 13
F:\charmi_monani>
```

p23

Write a java method to display the middle character of a string:

Note:

A]: If the length of a string is even there will be two middle characters.

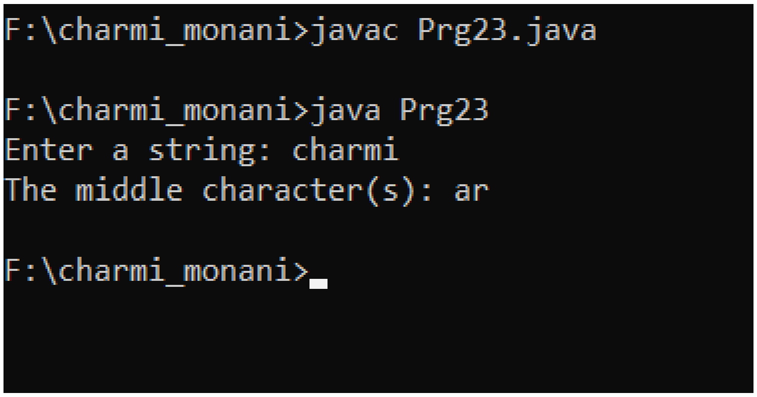
B]: If the length of a string is odd there will be one middle characters.

Soln:

```
import java.util.*;

class Prg23

{

    public static void main(String[] args)

            {
```

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

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

        String input = sc.nextLine();

        System.out.println("The middle character(s): " +
getMiddleCharacter(input));

    }


    public static String getMiddleCharacter(String str)

             {

        int length = str.length();

        int middle = length / 2;

        if (length % 2 == 0)

                     {

            // Even length: return two middle characters

            return str.substring(middle - 1, middle + 1);

        }

                     else

                     {

            // Odd length: return one middle character

            return str.substring(middle, middle + 1);

        }

    }

}
```

Output:



```
F:\charmi_monani>javac Prg23.java

F:\charmi_monani>java Prg23
Enter a string: charmi
The middle character(s): ar

F:\charmi_monani>
```

p24

Write a program to add and to multiply two int matrices.

Soln:

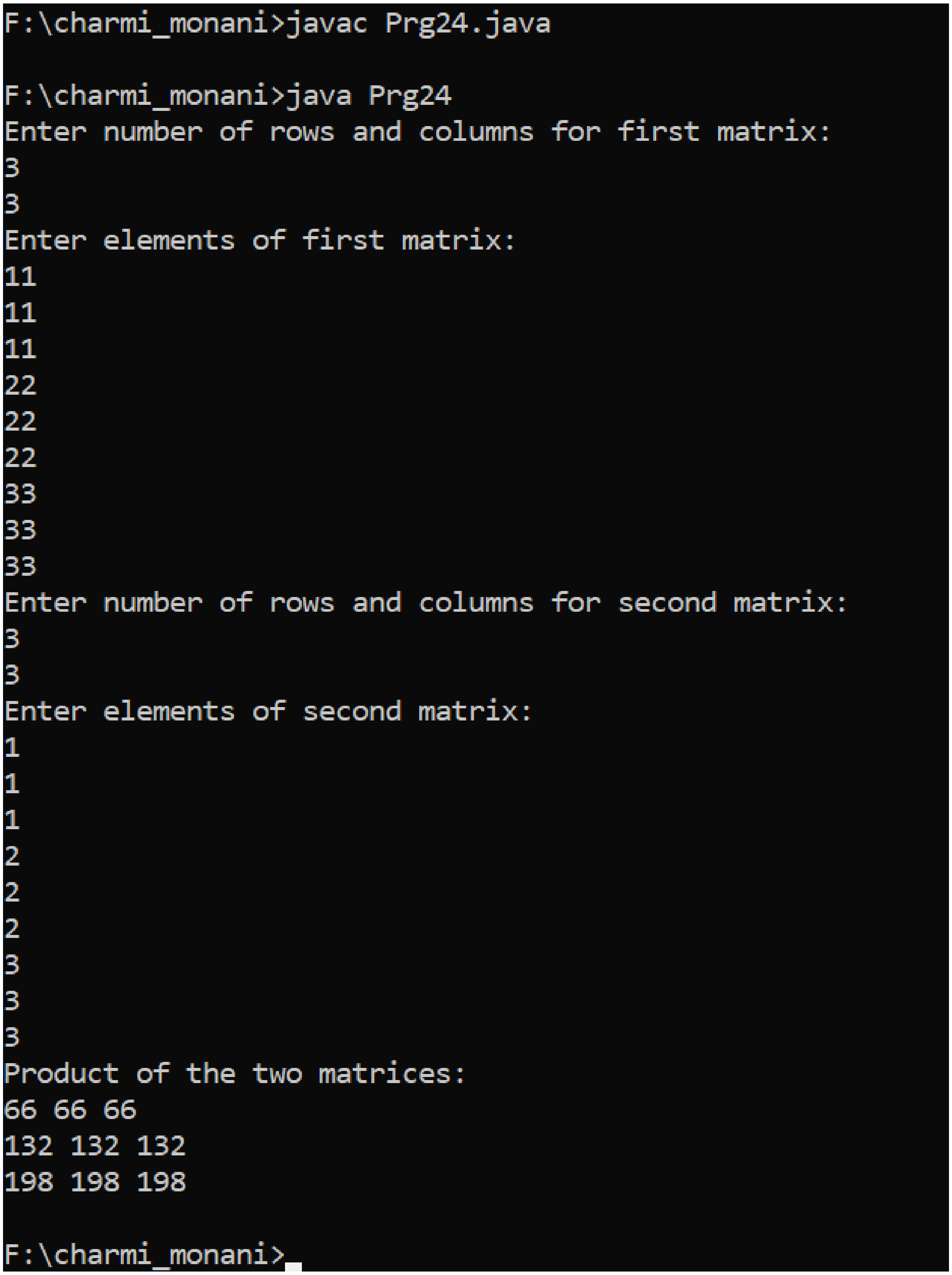
```
import java.util.*;

class Prg24

{

  public static void main(String[] args)

        {

    Scanner s1 = new Scanner(System.in);

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

    int rows = s1.nextInt();

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

    int columns = s1.nextInt();
```

```
int[][] matrix1 = new int[rows][columns];
int[][] matrix2 = new int[rows][columns];

System.out.println("Enter elements of the first matrix:");
for (int i = 0; i < rows; i++)
          {
  for (int j = 0; j < columns; j++)
                        {
    matrix1[i][j] = s1.nextInt();
  }
}
System.out.println("Enter elements of the second matrix:");
for (int i = 0; i < rows; i++)
          {
  for (int j = 0; j < columns; j++)
                        {
    matrix2[i][j] = s1.nextInt();
  }
}
int[][] sum = new int[rows][columns];
for (int i = 0; i < rows; i++)
          {
```

```
   for (int j = 0; j < columns; j++)
                                  {
      sum[i][j] = matrix1[i][j] + matrix2[i][j];
   }
}
int[][] mul = new int[rows][columns];
for (int i = 0; i < rows; i++)
              {
   for (int j = 0; j < columns; j++)
                                  {
      mul[i][j] = 0;
      for (int k = 0; k < columns; k++)
                                            {
         mul[i][j] += matrix1[i][k] * matrix2[k][j];
      }
   }
}
System.out.println("Sum of the matrices:");
for (int i = 0; i < rows; i++)
              {
   for (int j = 0; j < columns; j++)
                                  {
```

```
        System.out.print(sum[i][j] + " ");
      }
      System.out.println();
    }
    System.out.println("multiply of the matrices:");
    for (int i = 0; i < rows; i++)
              {
      for (int j = 0; j < columns; j++)
                            {
        System.out.print(mul[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```

Output:



```
F:\charmi_monani>javac Prg24.java

F:\charmi_monani>java Prg24
Enter number of rows and columns for first matrix:
3
3
Enter elements of first matrix:
11
11
11
22
22
22
33
33
33
Enter number of rows and columns for second matrix:
3
3
Enter elements of second matrix:
1
1
1
2
2
2
3
3
3
Product of the two matrices:
66 66 66
132 132 132
198 198 198

F:\charmi_monani>
```

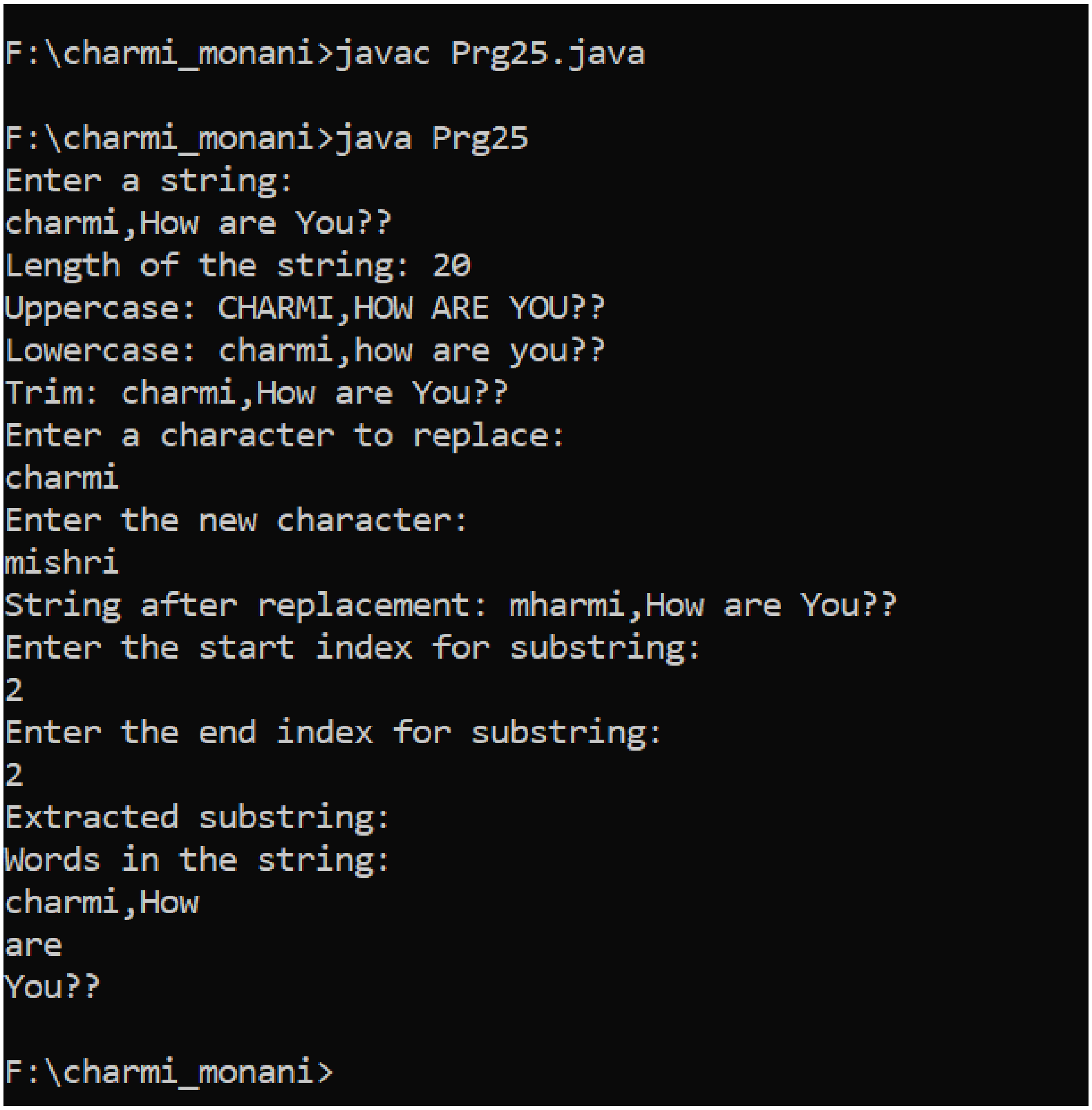
p25	**Write a program to implement all String operations.**

Soln:

```
// Perform all operations on the string.
import java.util.*;
public class Prg25
{
  public static void main(String[] args)
       {
    Scanner s1 = new Scanner(System.in);
    System.out.println("Enter a string:");
    String a = s1.nextLine();
    System.out.println("Length of the string: " + a.length());
    System.out.println("Uppercase: " + a.toUpperCase());
    System.out.println("Lowercase: " + a.toLowerCase());
              System.out.println("Trim: "+a.trim());
    System.out.println("Enter a character to replace:");
    char oldChar = s1.next().charAt(0);
    System.out.println("Enter the new character:");
    char newChar = s1.next().charAt(0);
    String replacedString = a.replace(oldChar, newChar);
    System.out.println("String after replacement: " + replacedString);
    System.out.println("Enter the start index for substring:");
```

```
        int startIndex = s1.nextInt();

        System.out.println("Enter the end index for substring:");

        int endIndex = s1.nextInt();

        String substring = a.substring(startIndex, endIndex);

        System.out.println("Extracted substring: " + substring);

        String[] words = a.split(" ");

        System.out.println("Words in the string:");

        for (String word : words)

                        {

           System.out.println(word);

        }

    }

}
```

Output:



```
F:\charmi_monani>javac Prg25.java

F:\charmi_monani>java Prg25
Enter a string:
charmi,How are You??
Length of the string: 20
Uppercase: CHARMI,HOW ARE YOU??
Lowercase: charmi,how are you??
Trim: charmi,How are You??
Enter a character to replace:
charmi
Enter the new character:
mishri
String after replacement: mharmi,How are You??
Enter the start index for substring:
2
Enter the end index for substring:
2
Extracted substring:
Words in the string:
charmi,How
are
You??

F:\charmi_monani>
```

Q.2) Programs for this,super,final and Static Keyword

(1)

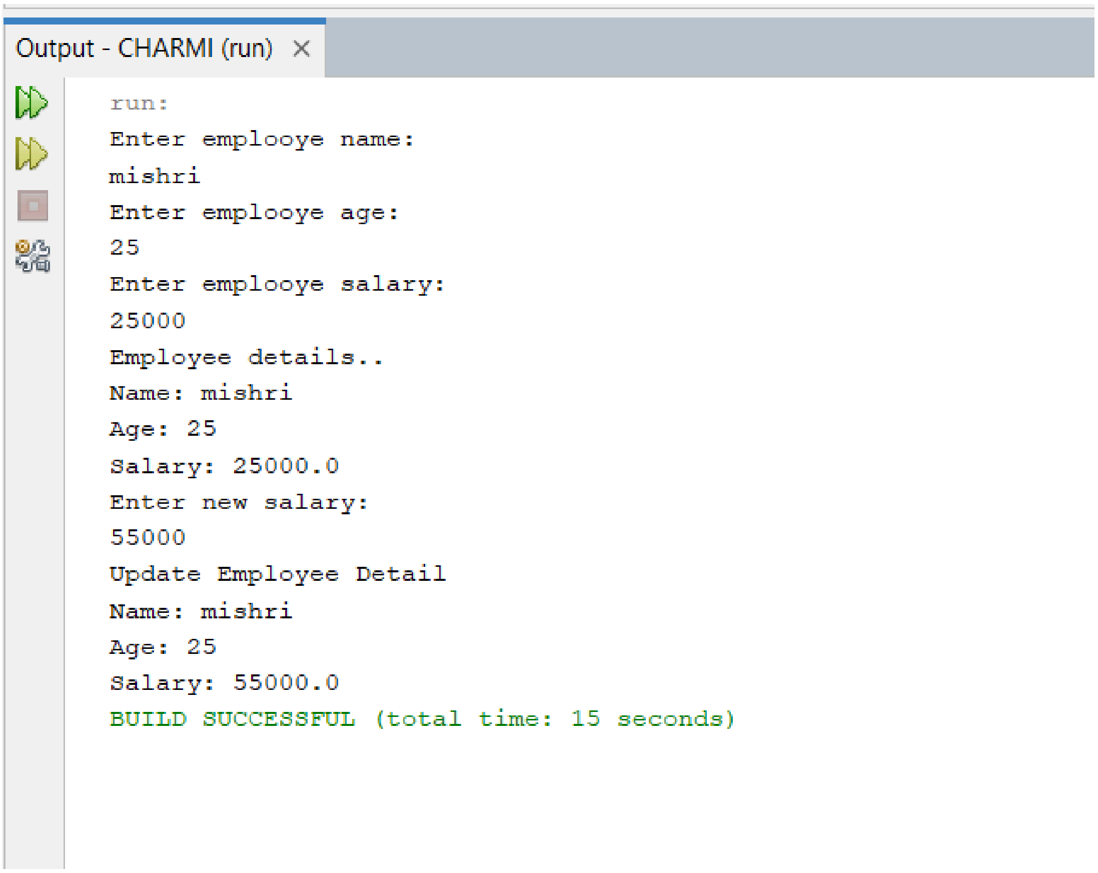
Write a program to show the concept of this keyword.

Soln:

// using this keyword....

```
import java.util.*;
class Employee
{
        private String name;
        private int age;
        private double salary;
        public Employee(String name, int age, double salary)
        {
                this.name=name;
                this.age=age;
                this.salary=salary;
        }
        public void display()
        {
                System.out.println("Name: "+this.name);
                System.out.println("Age: "+this.age);
                System.out.println("Salary: "+this.salary);
        }
        public void updataSalary(double newSalary)
        {
                this.salary=newSalary;
        }
}
 class Prg26
{
        public static void main(String args[])
        {
                Scanner s1=new Scanner(System.in);
                System.out.println("Enter emplooye name: ");
                String name=s1.next();
                System.out.println("Enter emplooye age: ");
                int age=s1.nextInt();
                System.out.println("Enter emplooye salary: ");
                double salary=s1.nextDouble();
                Employee e1=new Employee(name, age, salary);
                System.out.println("Employee details..");
                e1.display();
                System.out.println("Enter new salary: ");
                double newSalary=s1.nextDouble();
                e1.updataSalary(newSalary);
                System.out.println("Update Employee Detail");
                e1.display();
        }
}
```

Output:



```
run:
Enter emplooye name:
mishri
Enter emplooye age:
25
Enter emplooye salary:
25000
Employee details..
Name: mishri
Age: 25
Salary: 25000.0
Enter new salary:
55000
Update Employee Detail
Name: mishri
Age: 25
Salary: 55000.0
BUILD SUCCESSFUL (total time: 15 seconds)
```

(2)

Write a program to show the concept of super keyword.

Soln:

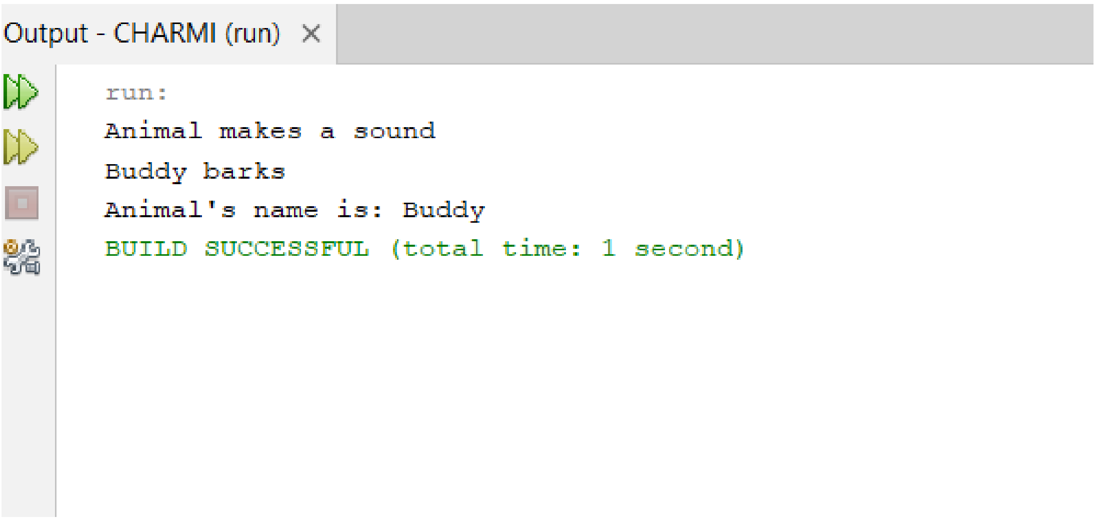
```
class Animal {
  String name;
```

```

    public Animal(String name) {

        this.name = name;

            }




    public void sound() {

        System.out.println("Animal makes a sound");

            }

}


// Child class that extends Animal

class Dog extends Animal {


            // Constructor of the child class

    public Dog(String name) {

        // Calling the constructor of the parent class using super

        super(name);

            }




    public void sound() {
```

```
    // Calling the parent class method using super

    super.sound();

    System.out.println(name + " barks");

      }


      // Method to display the name of the animal

  public void displayName() {

    // Accessing the parent class variable using super

    System.out.println("Animal's name is: " + super.name);

      }

}


public class pr27{

  public static void main(String[] args) {

    // Creating an object of Dog class

    Dog dog = new Dog("Buddy");


    // Calling the sound method which calls the parent class method too

    dog.sound();


    // Calling displayName method to show name of the animal from parent class

    dog.displayName();
```

```
        }
}
```

Output:



```
run:
Animal makes a sound
Buddy barks
Animal's name is: Buddy
BUILD SUCCESSFUL (total time: 1 second)
```

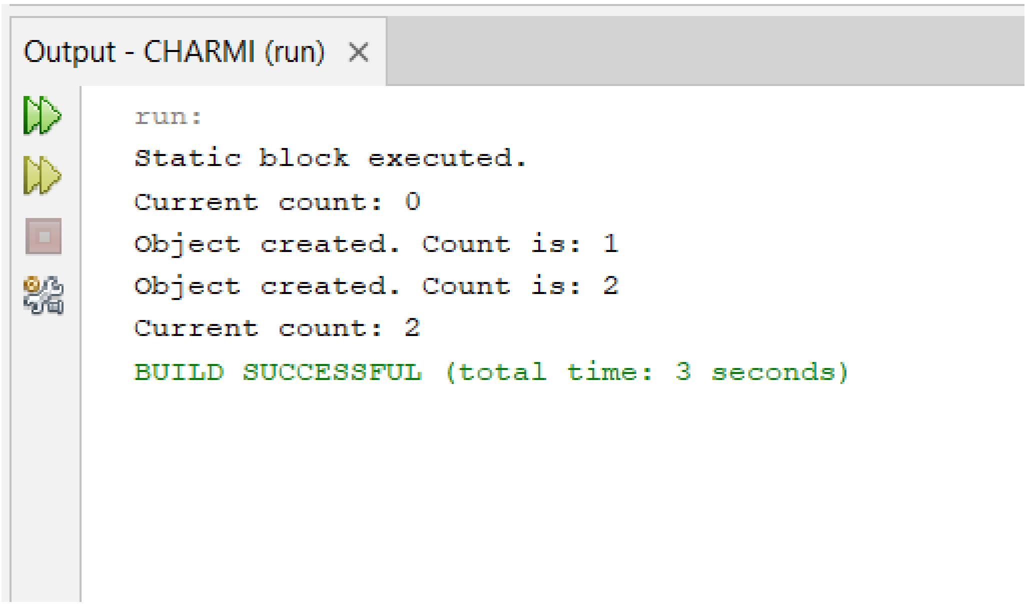
(3)

Write a program to show the concept of static keyword.

Soln:

```
class Counter {
        // Static variable to keep track of the count
  static int count = 0;

        // Constructor
  public Counter() {
      // Increment the count every time an object is created
      count++;
        }

        // Static method to display the value of the static variable
  public static void displayCount() {
      System.out.println("Current count: " + count);
        }

        // Non-static method
  public void showMessage() {
      System.out.println("Object created. Count is: " + count);
        }

        // Static block to initialize static variables
  static {
      System.out.println("Static block executed.");
        }
}

public class pr28 {
  public static void main(String[] args) {
      // Displaying the count without creating an object
      Counter.displayCount();


      // Creating first object of the Counter class
      Counter obj1 = new Counter();
      obj1.showMessage();


      // Creating second object of the Counter class
      Counter obj2 = new Counter();
      obj2.showMessage();


      // Displaying the count using the static method again
      Counter.displayCount();
        }
}
```

Output:



```
run:
Static block executed.
Current count: 0
Object created. Count is: 1
Object created. Count is: 2
Current count: 2
BUILD SUCCESSFUL (total time: 3 seconds)
```

(4)

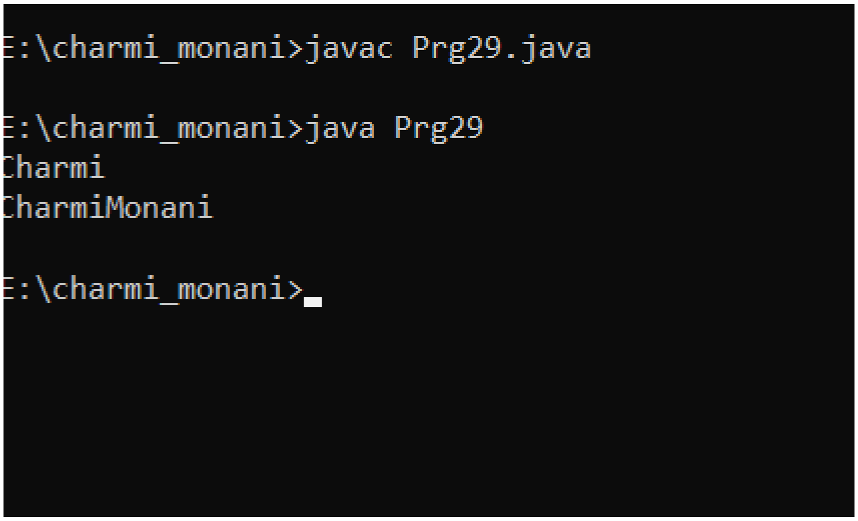
Write a program to show the concept of final keyword.

Soln:

```
class Prg29
{


  public static void main(String[] args)
  {

    final StringBuilder sb = new StringBuilder("Charmi");
```

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

    sb.append("Monani");


      System.out.println(sb);

  }

}
```

Output:



```
E:\charmi_monani>javac Prg29.java

E:\charmi_monani>java Prg29
Charmi
CharmiMonani

E:\charmi_monani>_
```

Q.3) Programs for Package, Constructor, polymorphism and Inheritance

(1)

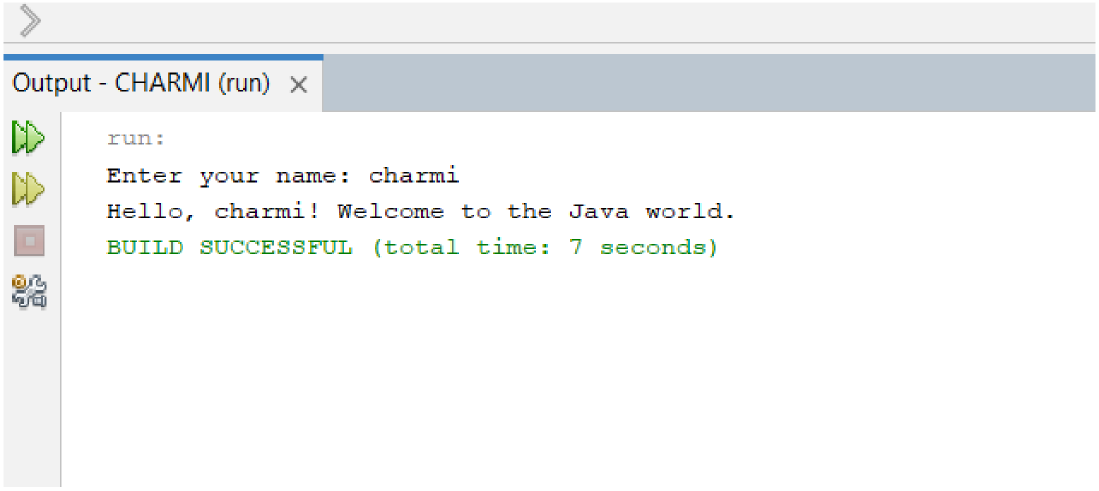
Write a program to show the concept of package.

Soln:

```
// Package declaration

package pack1;


import java.util.*;
```

```
class Prg30

{

   public static void main(String[] args)

          {

      // Create a Scanner object

      Scanner scanner = new Scanner(System.in);



      System.out.print("Enter your name: ");

      String name = scanner.nextLine();



      System.out.println("Hello, " + name + "! Welcome to the Java world.");

   }

}
```

Output:



```
run:
Enter your name: charmi
Hello, charmi! Welcome to the Java world.
BUILD SUCCESSFUL (total time: 7 seconds)
```

(2)

Create a class overload Demo and create method test() in it. Overload test() in four ways. First version takes no parameter, The second takes one integer parameter , and the third takes two integer parameter, and fourth takes one double parameter.

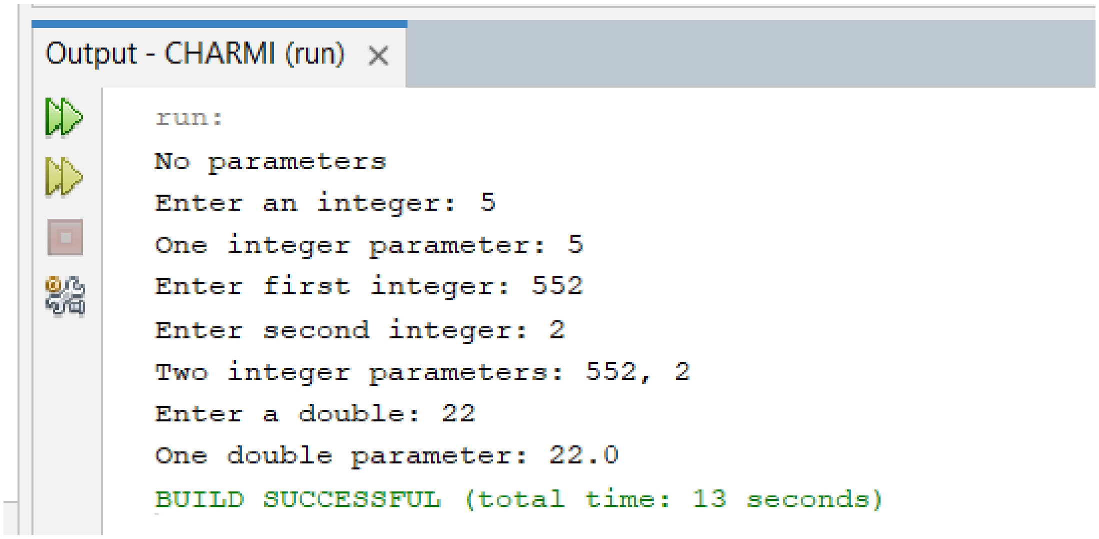
Soln:

```
import java.util.*;

// class Demo...

class Demo5

{

   // First version: no parameters

   public void test()

          {

      System.out.println("No parameters");

   }


   // Second version: one integer parameter

   public void test(int a)

          {

      System.out.println("One integer parameter: " + a);

   }
```

```
// Third version: two integer parameters

public void test(int a, int b)

        {

   System.out.println("Two integer parameters: " + a + ", " + b);

}


// Fourth version: one double parameter

public void test(double a)

        {

   System.out.println("One double parameter: " + a);

}


public static void main(String[] args)

        {

   Demo5 demo = new Demo5();

   Scanner scanner = new Scanner(System.in);


   // Using different overloaded methods

   demo.test();


   System.out.print("Enter an integer: ");

   int intVal = scanner.nextInt();
```

```
        demo.test(intVal);


        System.out.print("Enter first integer: ");
        int firstInt = scanner.nextInt();
        System.out.print("Enter second integer: ");
        int secondInt = scanner.nextInt();
        demo.test(firstInt, secondInt);


        System.out.print("Enter a double: ");
        double doubleVal = scanner.nextDouble();
        demo.test(doubleVal);
    }
}
```

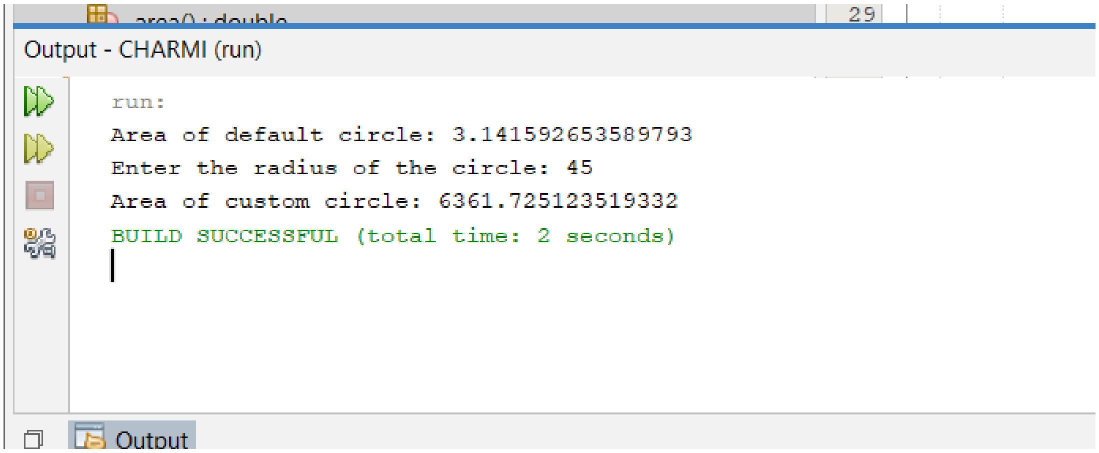
Output:

	
	Output - CHARMI (run)
	`run:`
	`No parameters`
	`Enter an integer: 5`
	`One integer parameter: 5`
	`Enter first integer: 552`
	`Enter second integer: 2`
	`Two integer parameters: 552, 2`
	`Enter a double: 22`
	`One double parameter: 22.0`
	`BUILD SUCCESSFUL (total time: 13 seconds)`

(3)	**Write a Java program which creates the Circle class and takes the two constructors of the Circle class. First constructor takes the default value for radius and second constructor takes radius as parameter. Create a method calcArea() and calculate the area of the Circle and print the area of the circle.**
	Soln:
	import java.util.Scanner;
	
	class Circle
	{
	  double radius;
	
	  Circle()
	      {
	    this.radius = 1.0;
	  }

```
    Circle(double radius)
            {
        this.radius = radius;
    }

    double area()
            {
        return Math.PI * radius * radius;
    }
}

public class pr32
{
    public static void main(String[] args)
            {
        Scanner s1 = new Scanner(System.in);



        Circle defaultCircle = new Circle();
        System.out.println("Area of default circle: " + defaultCircle.area());
        System.out.print("Enter the radius of the circle: ");

        double radius = s1.nextDouble();

        Circle customCircle = new Circle(radius);

        System.out.println("Area of custom circle: " + customCircle.area());

    }

}
```

Output:



```
run:
Area of default circle: 3.141592653589793
Enter the radius of the circle: 45
Area of custom circle: 6361.725123519332
BUILD SUCCESSFUL (total time: 2 seconds)
```

(4)

Create class ‘Point3D’. There are three constructors for ‘Point3D’. The first form accepts one double argument. The second form accepts two double arguments. The third form accepts three double arguments. These are used to initialize all of the instance variables. Also prints instance variables.

Soln:

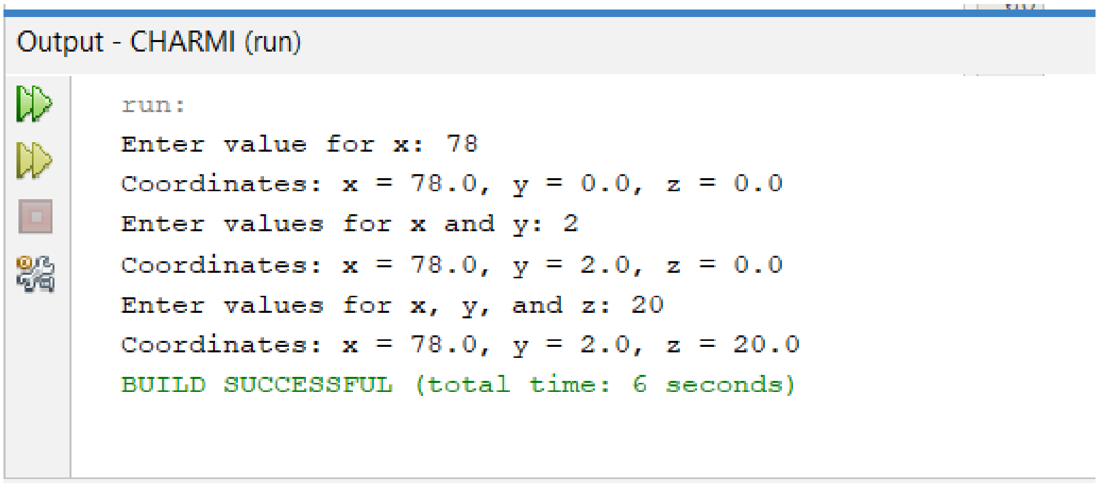
```
import java.util.*;

class Point3D
```

```
{
  private double x, y, z;

  // Constructor that accepts one double argument
  public Point3D(double x)
       {
    this.x = x;
    this.y = 0;
    this.z = 0;
    printCoordinates();
  }

  // Constructor that accepts two double arguments
  public Point3D(double x, double y)
       {
    this.x = x;
    this.y = y;
    this.z = 0;
    printCoordinates();
  }

  // Constructor that accepts three double arguments
```

```
public Point3D(double x, double y, double z)

     {

  this.x = x;

  this.y = y;

  this.z = z;

  printCoordinates();

}


// Method to print coordinates

private void printCoordinates()

     {

  System.out.println("Coordinates: x = " + x + ", y = " + y + ", z = " + z);

}


public static void main(String[] args)

     {

  Scanner scanner = new Scanner(System.in);


  // Test first constructor

  System.out.print("Enter value for x: ");

  double x = scanner.nextDouble();

  Point3D point1 = new Point3D(x);
```

```
        // Test second constructor

        System.out.print("Enter values for x and y: ");

        double y = scanner.nextDouble();

        Point3D point2 = new Point3D(x, y);


        // Test third constructor

        System.out.print("Enter values for x, y, and z: ");

        double z = scanner.nextDouble();

        Point3D point3 = new Point3D(x, y, z);

    }

}
```

Output:



Output - CHARMI (run)

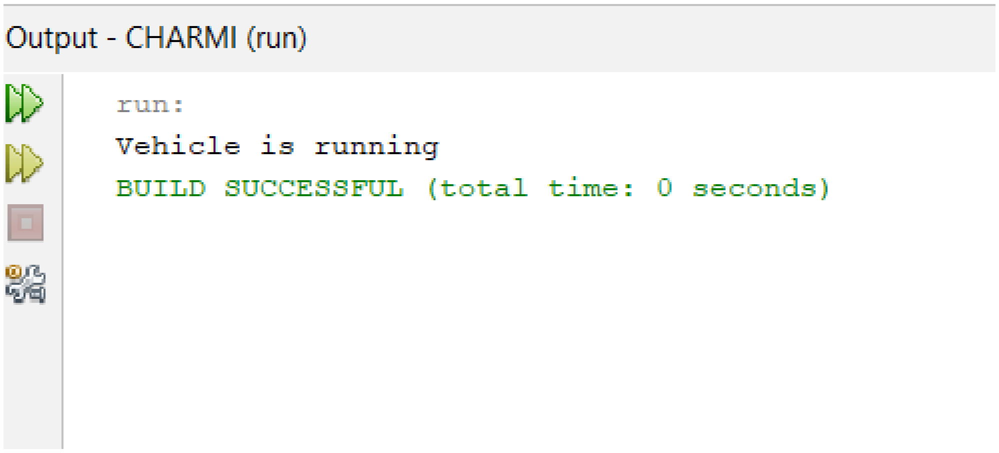
```
run:
Enter value for x: 78
Coordinates: x = 78.0, y = 0.0, z = 0.0
Enter values for x and y: 2
Coordinates: x = 78.0, y = 2.0, z = 0.0
Enter values for x, y, and z: 20
Coordinates: x = 78.0, y = 2.0, z = 20.0
BUILD SUCCESSFUL (total time: 6 seconds)
```

(5) **Write a program to show the concept of method overriding.**

Soln:

```
//Java Program to demonstrate why we need method overriding
//Here, we are calling the method of parent class with child
//class object.
//Creating a parent class
class Vehicle{
 void run(){System.out.println("Vehicle is running");}
}
//Creating a child class
class Bike extends Vehicle{
 public static void main(String args[]){
 //creating an instance of child class
 Bike obj = new Bike();
 //calling the method with child class instance
 obj.run();
 }
}
```

Output:



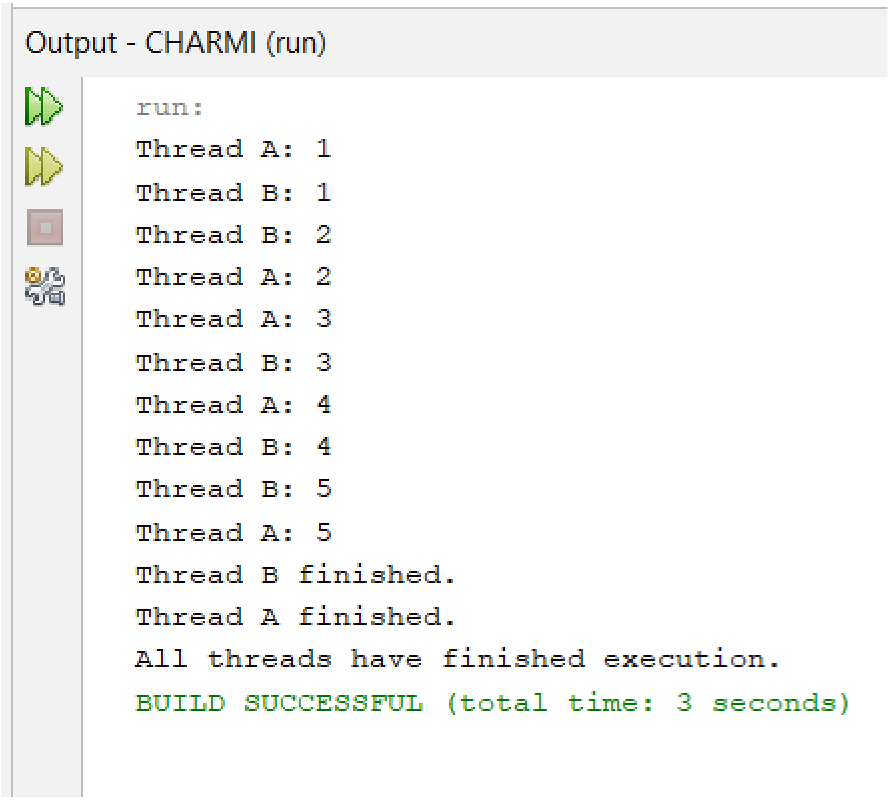
```
run:
Vehicle is running
BUILD SUCCESSFUL (total time: 0 seconds)
```

(6) **Write a program to show the concept of threading using Thread Class**

Soln:

```
// Demonstrating the concept of threading using the Thread class
public class ThreadExample {

    // Custom thread class extending the Thread class
    static class MyThread extends Thread {
        private String threadName;
        // Constructor to set the thread name
        public MyThread(String name) {
          this.threadName = name;
        }

        // Overriding the run() method to define the thread's task
        @Override
        public void run() {
          for (int i = 1; i <= 5; i++) {
            System.out.println(threadName + ": " + i);
            try {
              Thread.sleep(500); // Simulating some work with a 500ms delay
            } catch (InterruptedException e) {
              System.out.println(threadName + " interrupted.");
            }
          }
          System.out.println(threadName + " finished.");
        }
    }

    public static void main(String[] args) {
        // Creating thread instances
        MyThread thread1 = new MyThread("Thread A");
        MyThread thread2 = new MyThread("Thread B");

        // Starting threads
        thread1.start();
        thread2.start();

        // Waiting for threads to complete
        try {
          thread1.join();
          thread2.join();
        } catch (InterruptedException e) {
          System.out.println("Main thread interrupted.");
        }

        System.out.println("All threads have finished execution.");
    }
}
```

Output:



```
run:
Thread A: 1
Thread B: 1
Thread B: 2
Thread A: 2
Thread A: 3
Thread B: 3
Thread A: 4
Thread B: 4
Thread B: 5
Thread A: 5
Thread B finished.
Thread A finished.
All threads have finished execution.
BUILD SUCCESSFUL (total time: 3 seconds)
```

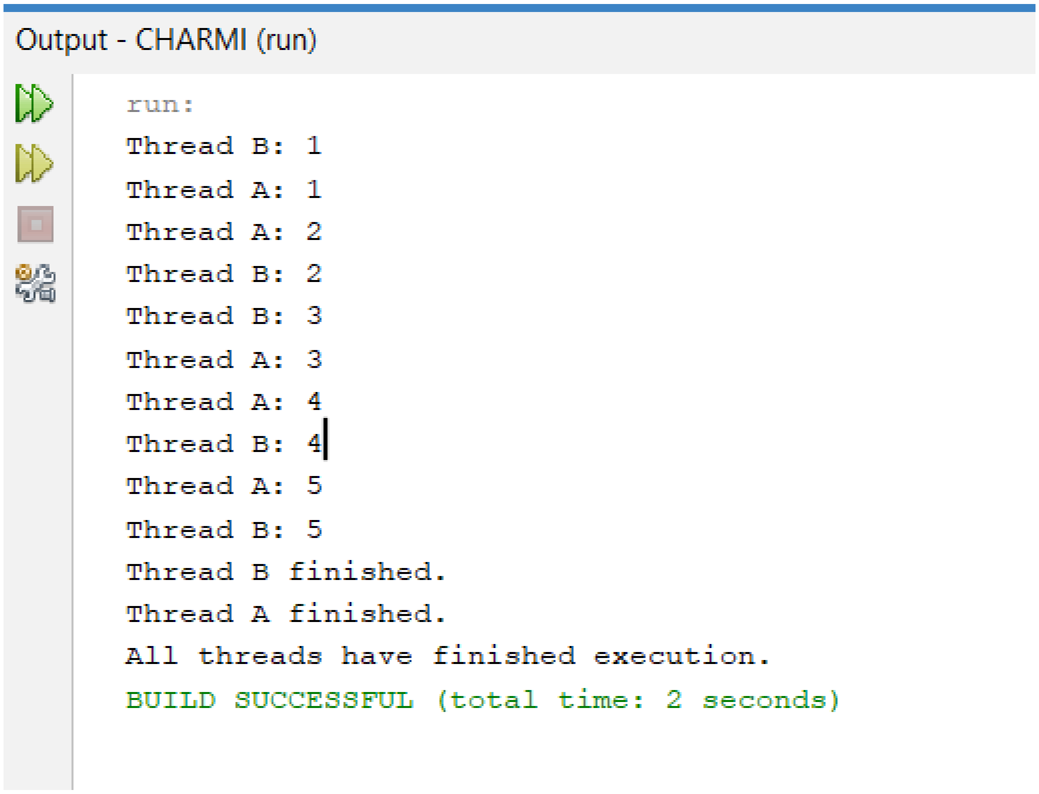
(5) **Write a program to show the concept of threading using runnable Interface.**

Soln:

```
public class RunnableExample {

  // Custom class implementing the Runnable interface
  static class MyRunnable implements Runnable {
    private String threadName;

    // Constructor to set the thread name
    public MyRunnable(String name) {
      this.threadName = name;
    }

    // Overriding the run() method to define the thread's task
    @Override
    public void run() {
      for (int i = 1; i <= 5; i++) {
        System.out.println(threadName + ": " + i);
        try {
          Thread.sleep(500); // Simulating some work with a 500ms delay
```

```
                } catch (InterruptedException e) {
                    System.out.println(threadName + " interrupted.");
                }
            }
            System.out.println(threadName + " finished.");
        }
    }

    public static void main(String[] args) {
        // Creating Runnable instances
        MyRunnable runnable1 = new MyRunnable("Thread A");
        MyRunnable runnable2 = new MyRunnable("Thread B");

        // Creating Thread objects and passing Runnable instances
        Thread thread1 = new Thread(runnable1);
        Thread thread2 = new Thread(runnable2);

        // Starting threads
        thread1.start();
        thread2.start();

        // Waiting for threads to complete
        try {
            thread1.join();
            thread2.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        System.out.println("All threads have finished execution.");
    }
}
```

Output:



Output - CHARMI (run)

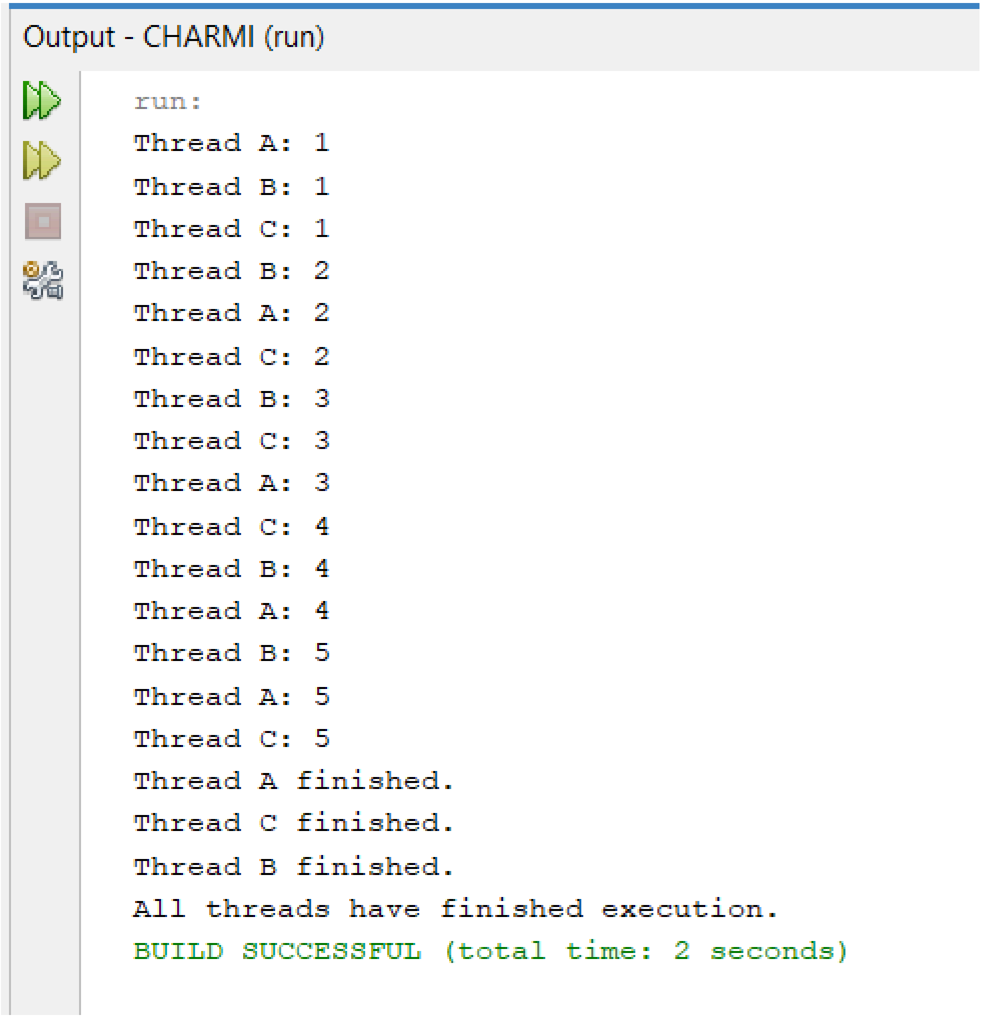
```
run:
Thread B: 1
Thread A: 1
Thread A: 2
Thread B: 2
Thread B: 3
Thread A: 3
Thread A: 4
Thread B: 4
Thread A: 5
Thread B: 5
Thread B finished.
Thread A finished.
All threads have finished execution.
BUILD SUCCESSFUL (total time: 2 seconds)
```

(6) **Write a program to implement thread priorities.**

Soln:

```
public class ThreadPriorityExample {

    // Custom thread class extending Thread
    static class MyThread extends Thread {
        public MyThread(String name) {
            super(name); // Set the thread name
        }

        @Override
        public void run() {
            for (int i = 1; i <= 5; i++) {
                System.out.println(getName() + ": " + i);
                try {
                    Thread.sleep(500); // Simulating some work with a 500ms delay
                } catch (InterruptedException e) {
                    System.out.println(getName() + " interrupted.");
                }
            }
            System.out.println(getName() + " finished.");
        }
    }

    public static void main(String[] args) {
        // Creating thread instances
        MyThread thread1 = new MyThread("Thread A");
        MyThread thread2 = new MyThread("Thread B");
        MyThread thread3 = new MyThread("Thread C");

        // Setting thread priorities
        thread1.setPriority(Thread.MIN_PRIORITY); // Priority 1
        thread2.setPriority(Thread.NORM_PRIORITY); // Priority 5 (default)
        thread3.setPriority(Thread.MAX_PRIORITY); // Priority 10

        // Starting threads
        thread1.start();
        thread2.start();
        thread3.start();

        // Waiting for threads to complete
        try {
            thread1.join();
            thread2.join();
            thread3.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        System.out.println("All threads have finished execution.");
    }
}
```

Output:



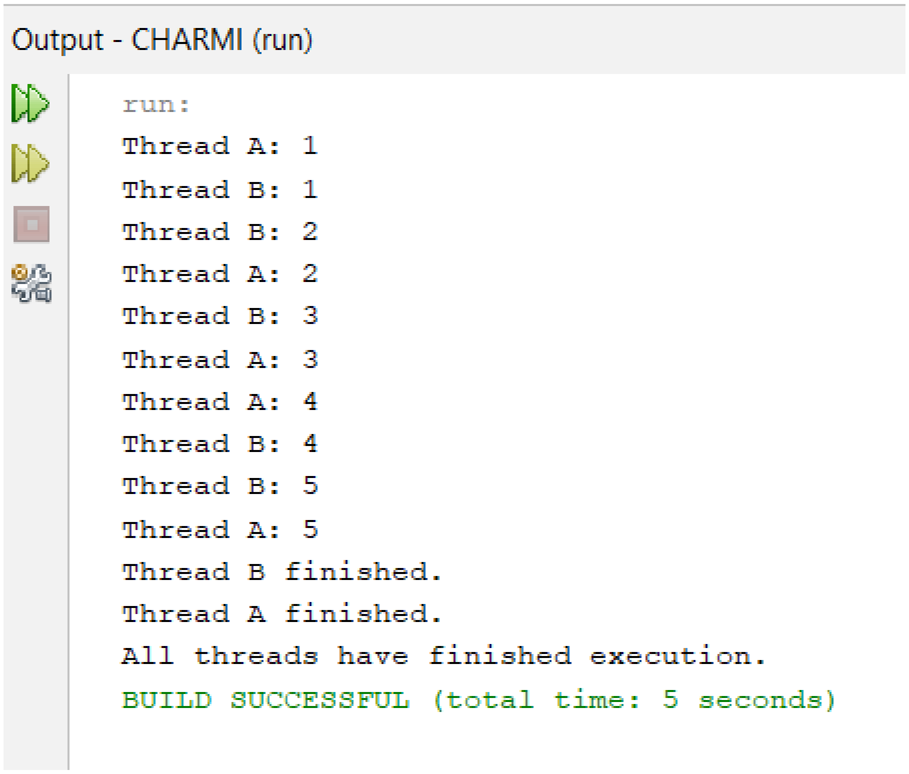
```
run:
Thread A: 1
Thread B: 1
Thread C: 1
Thread B: 2
Thread A: 2
Thread C: 2
Thread B: 3
Thread C: 3
Thread A: 3
Thread C: 4
Thread B: 4
Thread A: 4
Thread B: 5
Thread A: 5
Thread C: 5
Thread A finished.
Thread C finished.
Thread B finished.
All threads have finished execution.
BUILD SUCCESSFUL (total time: 2 seconds)
```

(7) **Write a program to show the concept of sleep method.in java**

Soln:

```
// Demonstrating the sleep() method in Java
public class SleepMethodExample {

    // Custom thread class extending Thread
    static class MyThread extends Thread {
        private String threadName;

        // Constructor to set the thread name
        public MyThread(String name) {
            this.threadName = name;
        }

        @Override
        public void run() {
            for (int i = 1; i <= 5; i++) {
                System.out.println(threadName + ": " + i);
                try {
                    // Pause the thread for 1 second (1000 milliseconds)
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    System.out.println(threadName + " interrupted.");
                }
            }
            System.out.println(threadName + " finished.");
        }
    }

    public static void main(String[] args) {
        // Creating thread instances
        MyThread thread1 = new MyThread("Thread A");
        MyThread thread2 = new MyThread("Thread B");

        // Starting threads
        thread1.start();
        thread2.start();

        // Waiting for threads to complete
        try {
            thread1.join();
            thread2.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        System.out.println("All threads have finished execution.");
    }
}
```

Output:



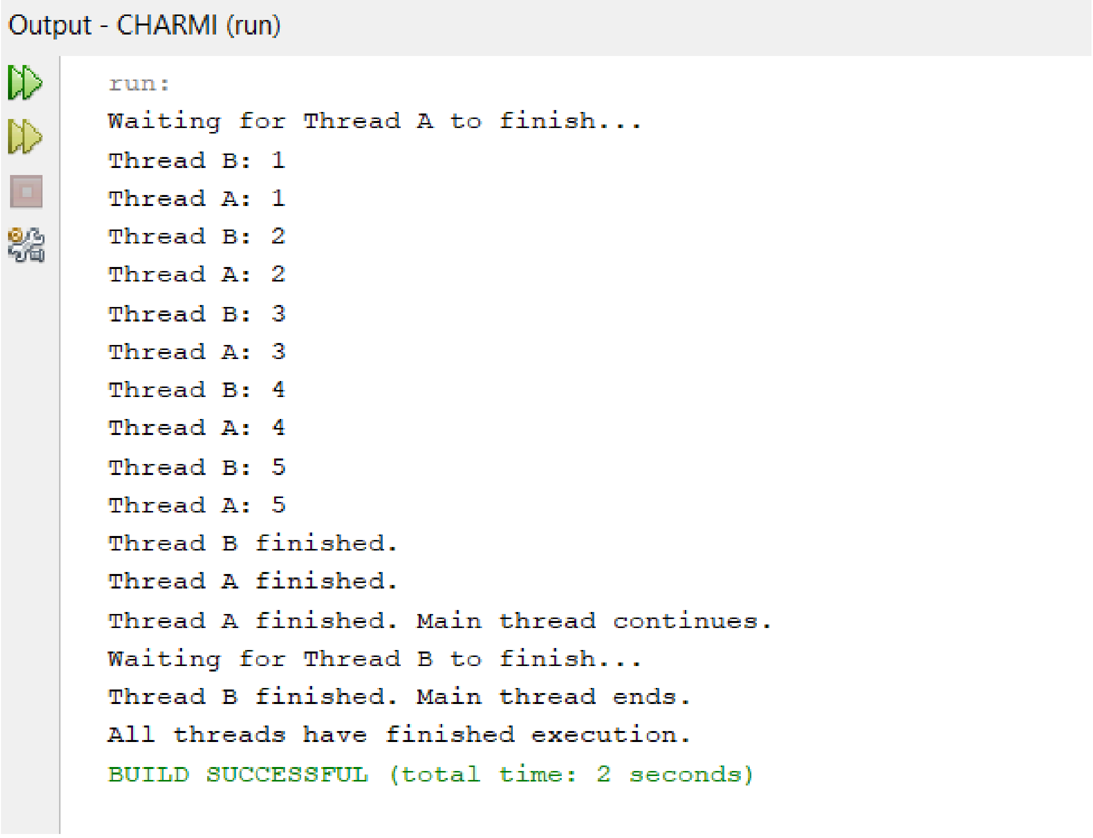
```
run:
Thread A: 1
Thread B: 1
Thread B: 2
Thread A: 2
Thread B: 3
Thread A: 3
Thread A: 4
Thread B: 4
Thread B: 5
Thread A: 5
Thread B finished.
Thread A finished.
All threads have finished execution.
BUILD SUCCESSFUL (total time: 5 seconds)
```

(8) **Write a program to show the concept of join method**

Soln:

```
public class JoinMethodExample {

   // Custom thread class extending Thread
   static class MyThread extends Thread {
      private String threadName;

      // Constructor to set the thread name
      public MyThread(String name) {
         this.threadName = name;
      }

      @Override
      public void run() {
         for (int i = 1; i <= 5; i++) {
            System.out.println(threadName + ": " + i);
            try {
               Thread.sleep(500); // Simulating some work with a 500ms delay
            } catch (InterruptedException e) {
               System.out.println(threadName + " interrupted.");
            }
         }
         System.out.println(threadName + " finished.");
      }
   }

   public static void main(String[] args) {
      // Creating thread instances
      MyThread thread1 = new MyThread("Thread A");
      MyThread thread2 = new MyThread("Thread B");

      // Starting threads
      thread1.start();
      thread2.start();

      // Using join() to wait for thread1 to finish
      try {
         System.out.println("Waiting for Thread A to finish...");
         thread1.join();
         System.out.println("Thread A finished. Main thread continues.");

         System.out.println("Waiting for Thread B to finish...");
         thread2.join();
         System.out.println("Thread B finished. Main thread ends.");
      } catch (InterruptedException e) {
         System.out.println("Main thread interrupted.");
      }

      System.out.println("All threads have finished execution.");
   }
}
```

Output:



```
run:
Waiting for Thread A to finish...
Thread B: 1
Thread A: 1
Thread B: 2
Thread A: 2
Thread B: 3
Thread A: 3
Thread B: 4
Thread A: 4
Thread B: 5
Thread A: 5
Thread B finished.
Thread A finished.
Thread A finished. Main thread continues.
Waiting for Thread B to finish...
Thread B finished. Main thread ends.
All threads have finished execution.
BUILD SUCCESSFUL (total time: 2 seconds)
```

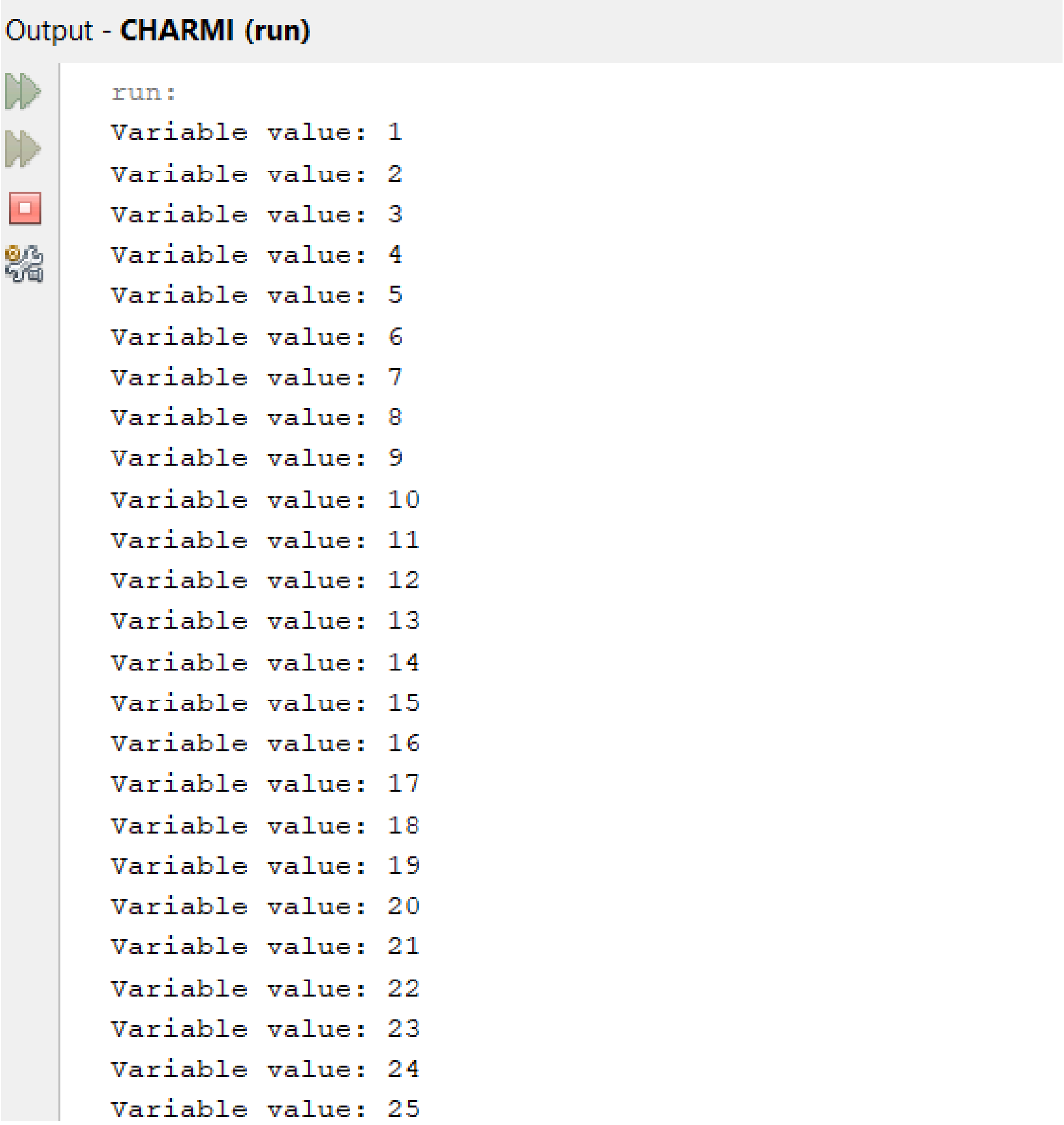
(9) **Write a program to increment the value of one variable by one and display it after one second using thread.**

Soln:

```
public class IncrementVariable {

  // Custom thread class extending Thread
  static class IncrementThread extends Thread {
    private int variable; // Variable to increment

    // Constructor to initialize the variable
    public IncrementThread(int initialValue) {
      this.variable = initialValue;
    }

    @Override
    public void run() {
      try {
        // Increment and display the variable every second
        while (true) {
          variable++;
```

```
                    System.out.println("Variable value: " + variable);
                    Thread.sleep(1000);
                }
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted.");
            }
        }
    }

    public static void main(String[] args) {
        // Initial value of the variable
        int initialValue = 0;
        IncrementThread incrementThread = new
IncrementThread(initialValue);
        incrementThread.start();
    }
}
```

Output:



Output - **CHARMI (run)**

```
run:
Variable value: 1
Variable value: 2
Variable value: 3
Variable value: 4
Variable value: 5
Variable value: 6
Variable value: 7
Variable value: 8
Variable value: 9
Variable value: 10
Variable value: 11
Variable value: 12
Variable value: 13
Variable value: 14
Variable value: 15
Variable value: 16
Variable value: 17
Variable value: 18
Variable value: 19
Variable value: 20
Variable value: 21
Variable value: 22
Variable value: 23
Variable value: 24
Variable value: 25
```

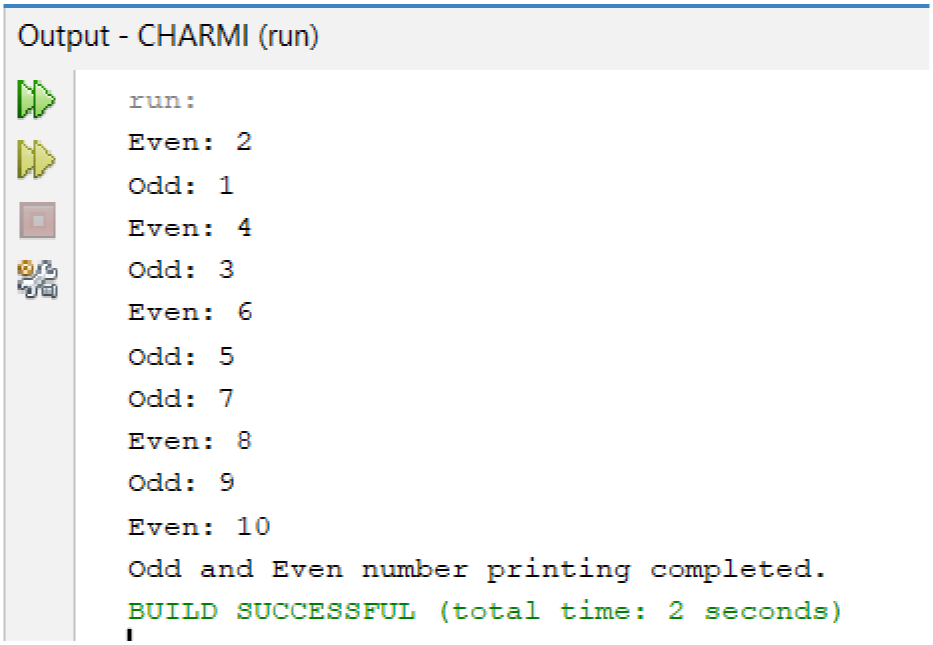
(10) **Write a program with 2 threads, in which one prints odd numbers and the other prints even numbers with delay of 500 milliseconds up to 10.**

Soln:

```
public class OddEvenThreadExample {

  // Thread class to print odd numbers
  static class OddThread extends Thread {
    @Override
    public void run() {
      for (int i = 1; i <= 10; i += 2) {
        System.out.println("Odd: " + i);
```

```
                try {
                    Thread.sleep(500); // Delay of 500 milliseconds
                } catch (InterruptedException e) {
                    System.out.println("OddThread interrupted.");
                }
            }
        }
    }

    // Thread class to print even numbers
    static class EvenThread extends Thread {
        @Override
        public void run() {
            for (int i = 2; i <= 10; i += 2) {
                System.out.println("Even: " + i);
                try {
                    Thread.sleep(500); // Delay of 500 milliseconds
                } catch (InterruptedException e) {
                    System.out.println("EvenThread interrupted.");
                }
            }
        }
    }

    public static void main(String[] args) {
        // Create instances of odd and even threads
        OddThread oddThread = new OddThread();
        EvenThread evenThread = new EvenThread();

        // Start both threads
        oddThread.start();
        evenThread.start();

        // Wait for both threads to complete
        try {
            oddThread.join();
            evenThread.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        System.out.println("Odd and Even number printing completed.");
    }
}
```

Output:



```
run:
Even: 2
Odd: 1
Even: 4
Odd: 3
Even: 6
Odd: 5
Odd: 7
Even: 8
Odd: 9
Even: 10
Odd and Even number printing completed.
BUILD SUCCESSFUL (total time: 2 seconds)
```

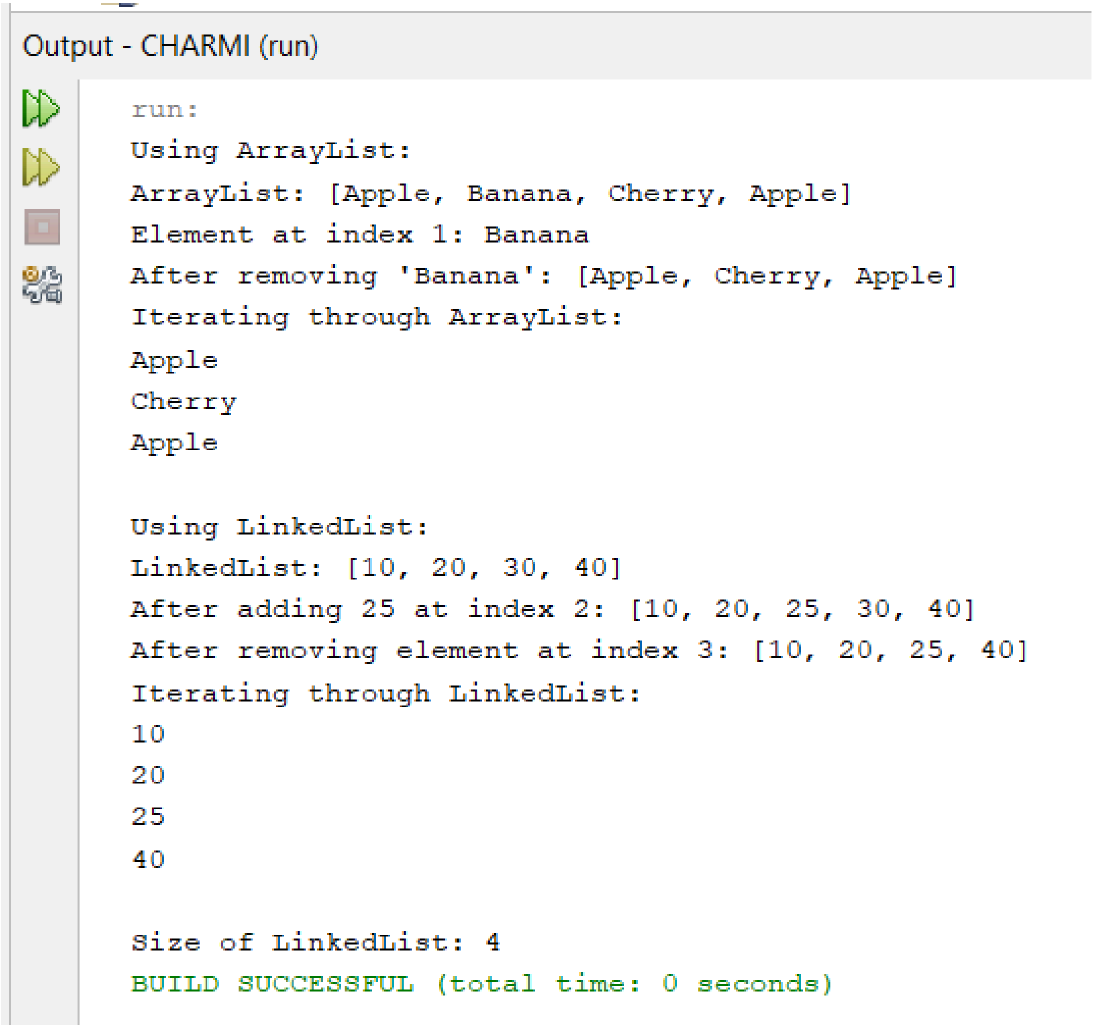
(11) **Write a program to demonstrate the use of the List Interface.**

Soln:

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

public class ListInterfaceExample {

  public static void main(String[] args) {
    // Using ArrayList (Dynamic Array Implementation)
    List<String> arrayList = new ArrayList<>();
    System.out.println("Using ArrayList:");
    arrayList.add("Apple");
    arrayList.add("Banana");
    arrayList.add("Cherry");
    arrayList.add("Apple"); // Duplicates allowed
    System.out.println("ArrayList: " + arrayList);

    // Accessing elements by index
    System.out.println("Element at index 1: " + arrayList.get(1));

    // Removing an element
    arrayList.remove("Banana");
    System.out.println("After removing 'Banana': " + arrayList);
```

```
        // Iterating through the list
        System.out.println("Iterating through ArrayList:");
        for (String fruit : arrayList) {
            System.out.println(fruit);
        }

        // Using LinkedList (Doubly-Linked List Implementation)
        List<Integer> linkedList = new LinkedList<>();
        System.out.println("\nUsing LinkedList:");
        linkedList.add(10);
        linkedList.add(20);
        linkedList.add(30);
        linkedList.add(40);
        System.out.println("LinkedList: " + linkedList);

        // Adding elements at a specific position
        linkedList.add(2, 25);
        System.out.println("After adding 25 at index 2: " + linkedList);

        // Removing elements by index
        linkedList.remove(3);
        System.out.println("After removing element at index 3: " + linkedList);

        // Iterating through the LinkedList
        System.out.println("Iterating through LinkedList:");
        for (Integer number : linkedList) {
            System.out.println(number);
        }

        // Checking the size of a list
        System.out.println("\nSize of LinkedList: " + linkedList.size());
    }
}
```

Output:



```
run:
Using ArrayList:
ArrayList: [Apple, Banana, Cherry, Apple]
Element at index 1: Banana
After removing 'Banana': [Apple, Cherry, Apple]
Iterating through ArrayList:
Apple
Cherry
Apple

Using LinkedList:
LinkedList: [10, 20, 30, 40]
After adding 25 at index 2: [10, 20, 25, 30, 40]
After removing element at index 3: [10, 20, 25, 40]
Iterating through LinkedList:
10
20
25
40

Size of LinkedList: 4
BUILD SUCCESSFUL (total time: 0 seconds)
```

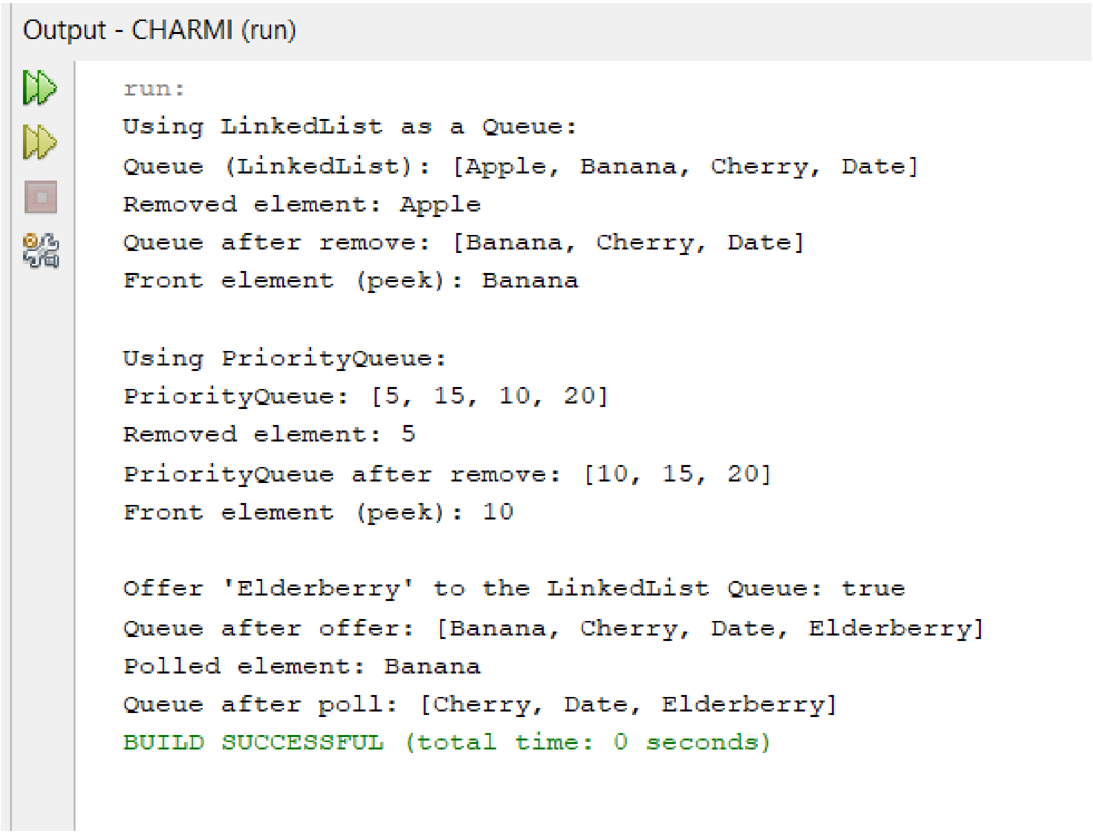
12

Write a program to demonstrate the use of the Queue Interface.

Soln:

```
import java.util.LinkedList;
import java.util.PriorityQueue;
import java.util.Queue;

public class QueueInterfaceExample {

  public static void main(String[] args) {
    // Using LinkedList as a Queue
    Queue<String> linkedListQueue = new LinkedList<>();
    System.out.println("Using LinkedList as a Queue:");
    linkedListQueue.add("Apple");
    linkedListQueue.add("Banana");
    linkedListQueue.add("Cherry");
```

```
        linkedListQueue.add("Date");

        // Printing the initial Queue
        System.out.println("Queue (LinkedList): " + linkedListQueue);

        // Removing elements from the front (FIFO order)
        System.out.println("Removed element: " + linkedListQueue.remove());
        System.out.println("Queue after remove: " + linkedListQueue);

        // Peek at the front element without removing it
        System.out.println("Front element (peek): " +
linkedListQueue.peek());

        // Using PriorityQueue (elements sorted in natural order)
        Queue<Integer> priorityQueue = new PriorityQueue<>();
        System.out.println("\nUsing PriorityQueue:");
        priorityQueue.add(10);
        priorityQueue.add(20);
        priorityQueue.add(5);
        priorityQueue.add(15);

        // Printing the initial PriorityQueue (elements are sorted)
        System.out.println("PriorityQueue: " + priorityQueue);

        // Removing elements (elements will be removed in sorted order)
        System.out.println("Removed element: " + priorityQueue.remove());
        System.out.println("PriorityQueue after remove: " + priorityQueue);

        // Peek at the front element (the smallest element in PriorityQueue)
        System.out.println("Front element (peek): " + priorityQueue.peek());

        // Using offer() method to add elements to the Queue
        boolean isAdded = linkedListQueue.offer("Elderberry");
        System.out.println("\nOffer 'Elderberry' to the LinkedList Queue: " +
isAdded);
        System.out.println("Queue after offer: " + linkedListQueue);

        // Using poll() method to remove elements (returns null if empty)
        String polledElement = linkedListQueue.poll();
        System.out.println("Polled element: " + polledElement);
        System.out.println("Queue after poll: " + linkedListQueue);
    }
}
```



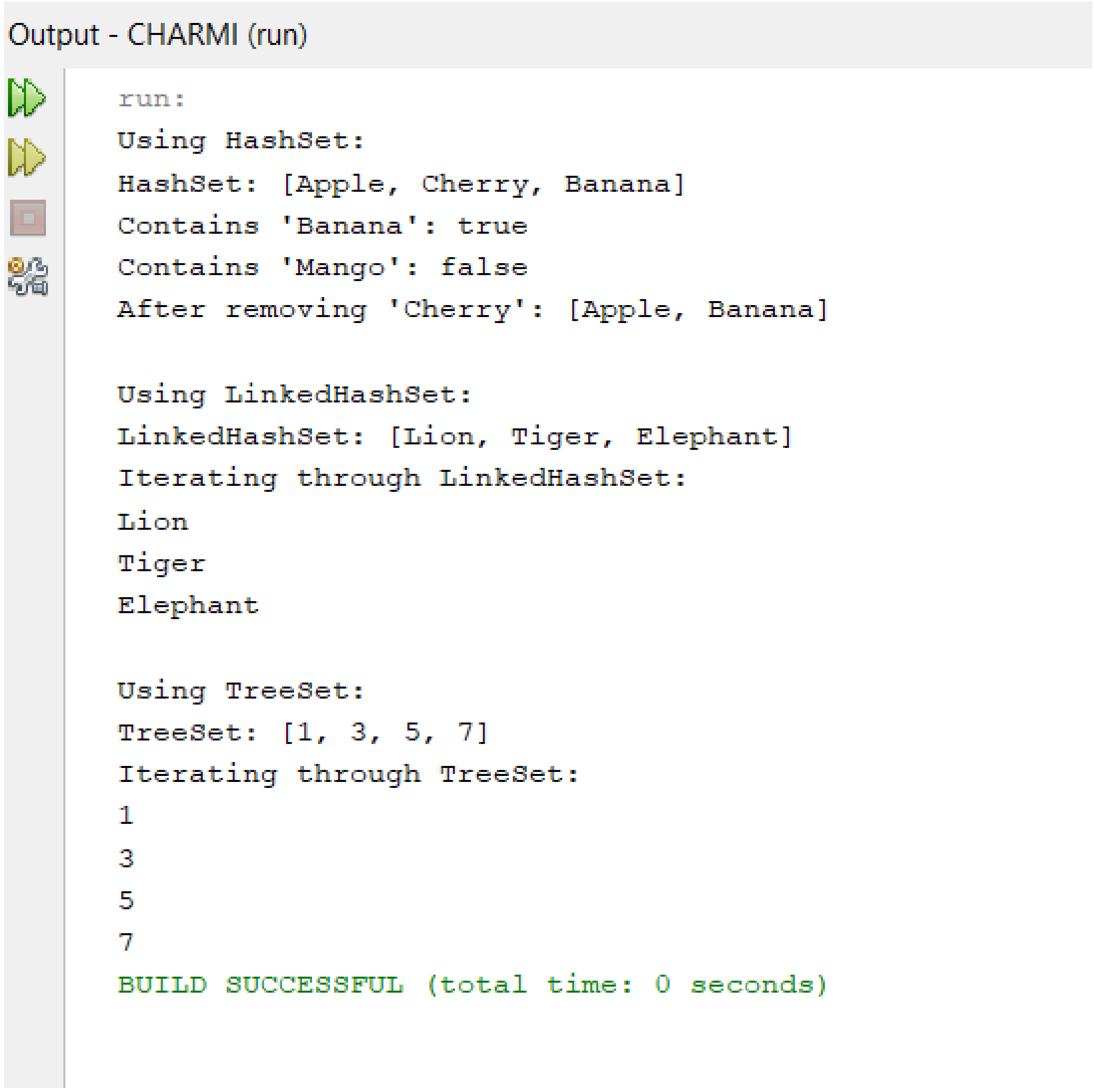
```
run:
Using LinkedList as a Queue:
Queue (LinkedList): [Apple, Banana, Cherry, Date]
Removed element: Apple
Queue after remove: [Banana, Cherry, Date]
Front element (peek): Banana

Using PriorityQueue:
PriorityQueue: [5, 15, 10, 20]
Removed element: 5
PriorityQueue after remove: [10, 15, 20]
Front element (peek): 10

Offer 'Elderberry' to the LinkedList Queue: true
Queue after offer: [Banana, Cherry, Date, Elderberry]
Polled element: Banana
Queue after poll: [Cherry, Date, Elderberry]
BUILD SUCCESSFUL (total time: 0 seconds)
```

(13) **Write a program to demonstrate the use of the Set Interface**

Soln:

```
import java.util.HashSet;
import java.util.LinkedHashSet;
import java.util.TreeSet;
import java.util.Set;

public class SetInterfaceExample {

    public static void main(String[] args) {
        // Using HashSet (Unordered Set with no duplicates)
        Set<String> hashSet = new HashSet<>();
        System.out.println("Using HashSet:");
        hashSet.add("Apple");
        hashSet.add("Banana");
        hashSet.add("Cherry");
        hashSet.add("Apple"); // Duplicate element
        System.out.println("HashSet: " + hashSet); // Duplicates will not be shown

        // Checking if an element exists in the set
        System.out.println("Contains 'Banana': " +
hashSet.contains("Banana"));
        System.out.println("Contains 'Mango': " + hashSet.contains("Mango"));

        // Removing an element from the set
        hashSet.remove("Cherry");
        System.out.println("After removing 'Cherry': " + hashSet);

        // Using LinkedHashSet (Maintains insertion order)
        Set<String> linkedHashSet = new LinkedHashSet<>();
        System.out.println("\nUsing LinkedHashSet:");
        linkedHashSet.add("Lion");
        linkedHashSet.add("Tiger");
        linkedHashSet.add("Elephant");
        linkedHashSet.add("Lion"); // Duplicate element
        System.out.println("LinkedHashSet: " + linkedHashSet); // Duplicates
will not be shown

        // Iterating through the LinkedHashSet
        System.out.println("Iterating through LinkedHashSet:");
        for (String animal : linkedHashSet) {
            System.out.println(animal);
        }

        // Using TreeSet (Sorted Set, ordered in ascending order)
        Set<Integer> treeSet = new TreeSet<>();
        System.out.println("\nUsing TreeSet:");
        treeSet.add(5);
        treeSet.add(3);
        treeSet.add(7);
        treeSet.add(1);
        treeSet.add(3); // Duplicate element
        System.out.println("TreeSet: " + treeSet); // Duplicates will not be
shown

        // Iterating through the TreeSet
        System.out.println("Iterating through TreeSet:");
        for (Integer number : treeSet) {
            System.out.println(number);
        }
    }
}
```

Output:



Output - CHARMI (run)

```
run:
Using HashSet:
HashSet: [Apple, Cherry, Banana]
Contains 'Banana': true
Contains 'Mango': false
After removing 'Cherry': [Apple, Banana]

Using LinkedHashSet:
LinkedHashSet: [Lion, Tiger, Elephant]
Iterating through LinkedHashSet:
Lion
Tiger
Elephant

Using TreeSet:
TreeSet: [1, 3, 5, 7]
Iterating through TreeSet:
1
3
5
7
BUILD SUCCESSFUL (total time: 0 seconds)
```

Q.5) Program for Swing

(1) **Create a simple window with a label that displays "Your Full Name!".**

(2) **Create a login form where the user inputs a username and password. Display a success message (Login Successfully) if the credentials are correct.**

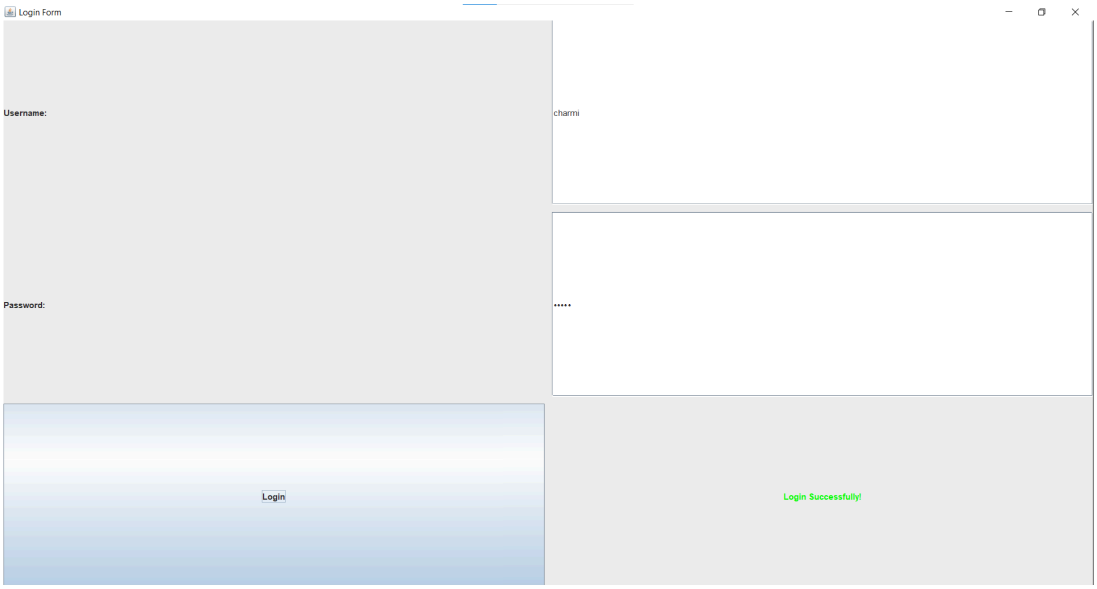
Soln:

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

```
public class LoginForm {
  public static void main(String[] args) {
    // Create the frame
    JFrame f = new JFrame("Login Form");
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setSize(350, 200);
    f.setLayout(new GridLayout(3, 2, 10, 10));


    String validUsername = "charmi";
    String validPassword = "12345";


    JLabel userLabel = new JLabel("Username:");
    JTextField userField = new JTextField();
    JLabel passLabel = new JLabel("Password:");
    JPasswordField passField = new JPasswordField();
    JButton loginButton = new JButton("Login");
    JLabel messageLabel = new JLabel("", SwingConstants.CENTER);


    loginButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        String username = userField.getText();
        String password = new String(passField.getPassword());

        if (username.equals(validUsername) &&
password.equals(validPassword)) {
          messageLabel.setText("Login Successfully!");
          messageLabel.setForeground(Color.GREEN);
        } else {
          messageLabel.setText("Invalid Credentials");
          messageLabel.setForeground(Color.RED);
        }
      }
    });


    f.add(userLabel);
    f.add(userField);
    f.add(passLabel);
    f.add(passField);
    f.add(loginButton);
    f.add(messageLabel);
```

```
        f.setVisible(true);
    }
}
```

Output:



Username: charmi

Password: •••••

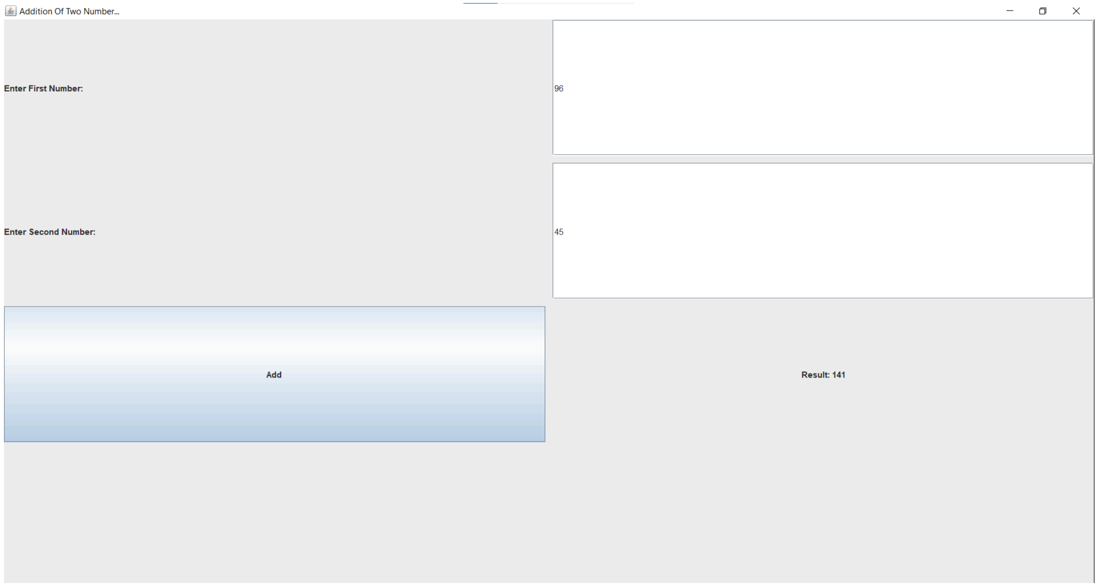
Login Successfully!

(3) Write a Java Program to create a Frame in swing that receives an integer in two text field, and computes its Addition value and returns it in Label, when the button named “Add” is clicked.

Soln:

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

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

        JFrame frame = new JFrame("Addition Of Two Number...");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);
        frame.setLayout(new GridLayout(4, 2, 10, 10));


        JLabel firstNumberLabel = new JLabel("Enter First Number:");
        JTextField firstNumberField = new JTextField();
        JLabel secondNumberLabel = new JLabel("Enter Second Number:");
        JTextField secondNumberField = new JTextField();
        JButton addButton = new JButton("Add");
    JLabel resultLabel = new JLabel("Result: ", SwingConstants.CENTER);


    addButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        try {
          // Parse integers from text fields
          int firstNumber = Integer.parseInt(firstNumberField.getText());
          int secondNumber =
Integer.parseInt(secondNumberField.getText());

          // Compute the sum
          int sum = firstNumber + secondNumber;

          // Display the result
          resultLabel.setText("Result: " + sum);
        } catch (NumberFormatException ex) {
          // Handle invalid input
          resultLabel.setText("Please enter valid integers!");
        }
      }
    });

    // Add components to the frame
    frame.add(firstNumberLabel);
    frame.add(firstNumberField);
    frame.add(secondNumberLabel);
    frame.add(secondNumberField);
    frame.add(addButton);
    frame.add(resultLabel);

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

Output:



Addition Of Two Number...

Enter First Number: 96

Enter Second Number: 45

Add

Result: 141

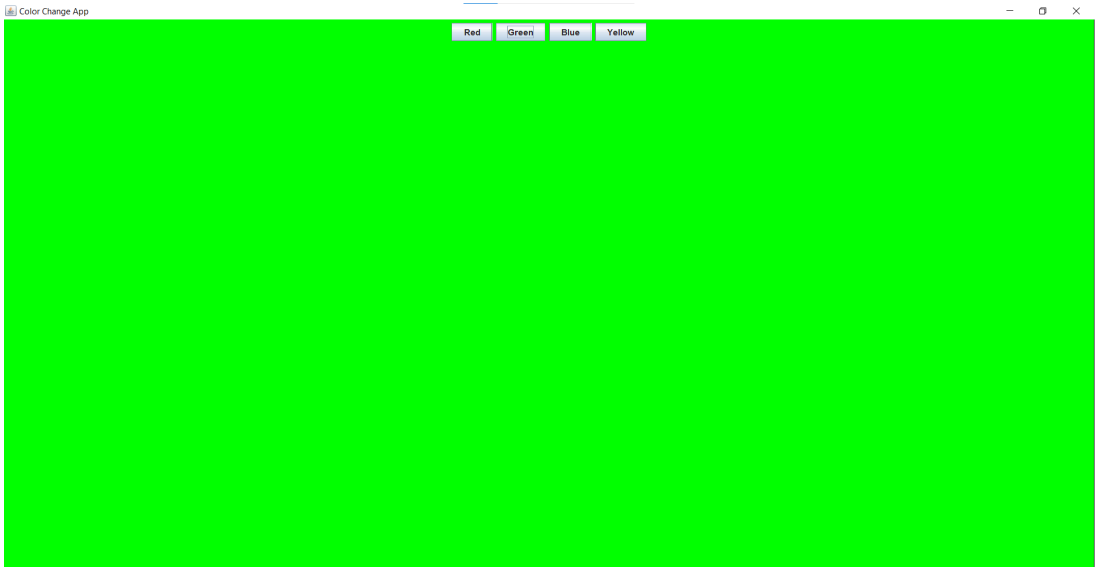
(4) **Write a Java program to create a Frame in swing that displays 4buttons each represents different colors.if a user click on particular button then that color is set as back ground to frame.**

Soln:

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

public class ColorChangerFrame {
  public static void main(String[] args) {
    // Create the frame
    JFrame frame = new JFrame("Color Change App");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 200);
    frame.setLayout(new FlowLayout());

    // Create buttons for different colors
    JButton redButton = new JButton("Red");
    JButton greenButton = new JButton("Green");
    JButton blueButton = new JButton("Blue");
    JButton yellowButton = new JButton("Yellow");

    // Add action listeners to change background color
    redButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        frame.getContentPane().setBackground(Color.RED);
```

```
            }
        });

        greenButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.getContentPane().setBackground(Color.GREEN);
            }
        });

        blueButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.getContentPane().setBackground(Color.BLUE);
            }
        });

        yellowButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                frame.getContentPane().setBackground(Color.YELLOW);
            }
        });

        // Add buttons to the frame
        frame.add(redButton);
        frame.add(greenButton);
        frame.add(blueButton);
        frame.add(yellowButton);

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

Output:

	Color Change App Red Green Blue Yellow