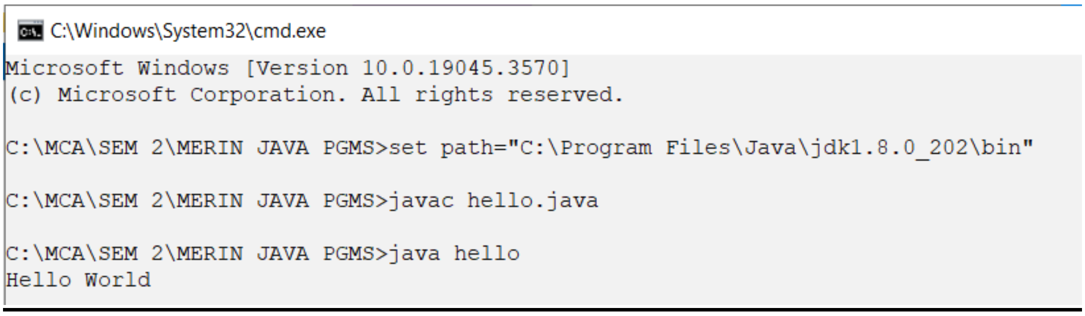
SECTION 1

1. Write a program to print ‘Welcome to Java’.

Source Code

```
import java.io.*;
public class hello
{
  public static void main(String args[])
   {
     System.out.println("Hello World");
   }
}
```

OUTPUT



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\MCA\SEM 2\MERIN JAVA PGMS>set path="C:\Program Files\Java\jdk1.8.0_202\bin"

C:\MCA\SEM 2\MERIN JAVA PGMS>javac hello.java

C:\MCA\SEM 2\MERIN JAVA PGMS>java hello
Hello World
```

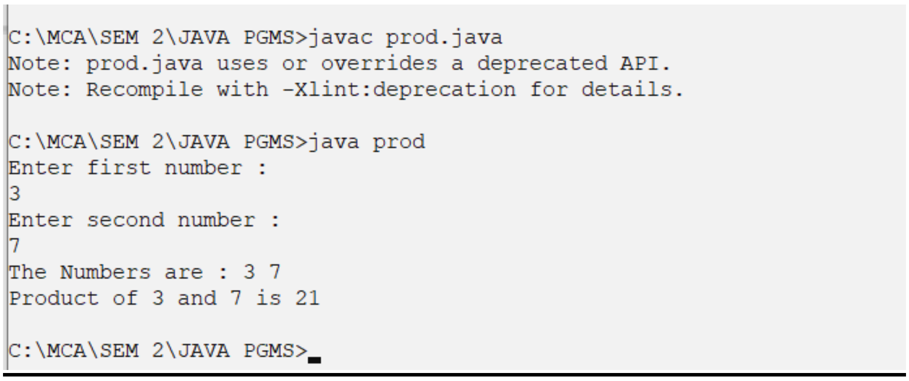
2. WAP to display two numbers received as command line argument, and print its product

SOURCE CODE

```
import java.io.*;
class prod
{
  public static void main(String args[])
   {
     DataInputStream din = new DataInputStream(System.in);
     int a,b;
     try
      {
        System.out.println("Enter first number : ");
        a=Integer.parseInt(din.readLine());
        System.out.println("Enter second number : ");
        b=Integer.parseInt(din.readLine());
        System.out.println("The Numbers are : "+a+" "+b);
```

```
            System.out.println("Product of "+a+" and "+b+" is "+ (a*b));
          }
        catch(Exception e)
          {
            System.out.println("Error : "+e);
          }
      }
  }
```

OUTPUT



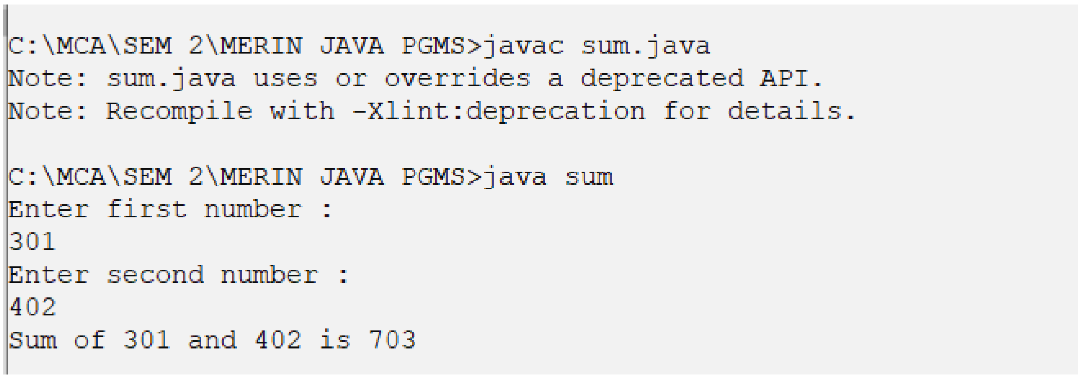
```
C:\MCA\SEM 2\JAVA PGMS>javac prod.java
Note: prod.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\JAVA PGMS>java prod
Enter first number :
3
Enter second number :
7
The Numbers are : 3 7
Product of 3 and 7 is 21

C:\MCA\SEM 2\JAVA PGMS>_
```

3. **WAP to read two numbers and display the output in the form of 'Sum of 2 and 3 is 5**

SOURCE CODE

```
import java.io.*;
class sum
{
   public static void main(String args[])
    {
      DataInputStream din = new DataInputStream(System.in);
      int a,b;
      try
       {
          System.out.println("Enter first number : ");
          a=Integer.parseInt(din.readLine());
          System.out.println("Enter second number : ");
          b=Integer.parseInt(din.readLine());
          System.out.println("Sum of "+a+" and "+b+" is "+ (a+b));
       }
      catch(Exception e)
       {
          System.out.println("Error : "+e);
       }
    }
}
```

OUTPUT



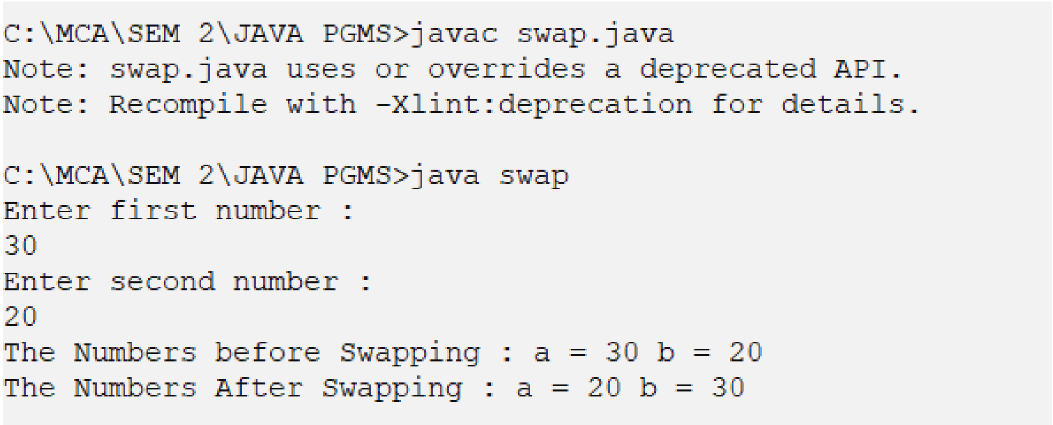
```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac sum.java
Note: sum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java sum
Enter first number :
301
Enter second number :
402
Sum of 301 and 402 is 703
```

4. WAP to accept two numbers from the keyboard and swap them.

SOURCE CODE

```
import java.io.*;
class swap
{
  public static void main(String args[])
   {
     DataInputStream din = new DataInputStream(System.in);
     int a,b,temp;
     try
      {
        System.out.println("Enter first number : ");
        a=Integer.parseInt(din.readLine());
        System.out.println("Enter second number : ");
        b=Integer.parseInt(din.readLine());
        System.out.println("The Numbers before Swapping : a = "+a+" b = "+b);
               temp=a;
               a=b;
               b=temp;
        System.out.println("The Numbers After Swapping : a = "+a+" b = "+b);
      }
     catch(Exception e)
      {
        System.out.println("Error : "+e);
      }
   }
}
```

OUTPUT



```
C:\MCA\SEM 2\JAVA PGMS>javac swap.java
Note: swap.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\JAVA PGMS>java swap
Enter first number :
30
Enter second number :
20
The Numbers before Swapping : a = 30 b = 20
The Numbers After Swapping : a = 20 b = 30
```

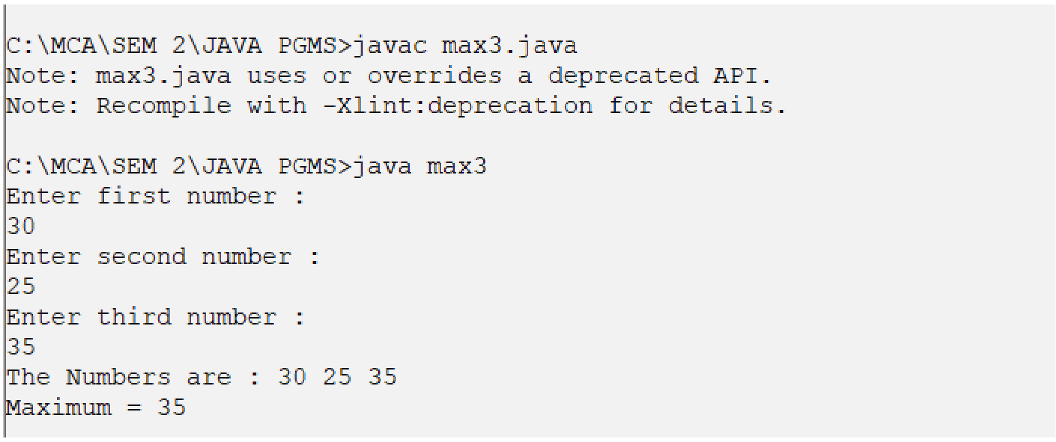
SECTION 2

1. **WAP to read three numbers and the maximum.**

Source Code

```
import java.io.*;
class max3
{
  public static void main(String args[])
   {
     DataInputStream din = new DataInputStream(System.in);
     int a,b,c;
     try
      {
        System.out.println("Enter first number : ");
        a=Integer.parseInt(din.readLine());
        System.out.println("Enter second number : ");
        b=Integer.parseInt(din.readLine());
              System.out.println("Enter third number : ");
        c=Integer.parseInt(din.readLine());
        System.out.println("The Numbers are : "+a+" "+b+" "+c);
              if(a>b)
              {
                      if(a>c)
                      {
                              System.out.println("Maximum = " +a);
                      }
                      else
                              System.out.println("Maximum = " +c);
              }
              else
                      if(b>c)
                              System.out.println("Maximum = " +b);
                      else
                              System.out.println("Maximum = " +c);
      }
     catch(Exception e)
      {
        System.out.println("Error : "+e);
      }
   }
}
```

OUTPUT



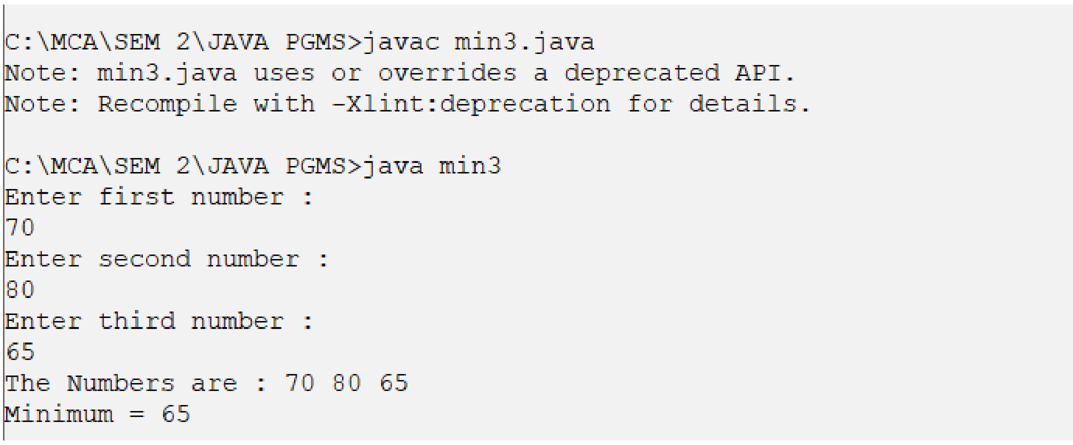
```
C:\MCA\SEM 2\JAVA PGMS>javac max3.java
Note: max3.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\JAVA PGMS>java max3
Enter first number :
30
Enter second number :
25
Enter third number :
35
The Numbers are : 30 25 35
Maximum = 35
```

2. **Find the minimum of three numbers using a single statement.**

SOURCE CODE

```
import java.io.*;
class min3
{
   public static void main(String args[])
    {
      DataInputStream din = new DataInputStream(System.in);
      int a,b,c,min;
      try
       {
          System.out.println("Enter first number : ");
          a=Integer.parseInt(din.readLine());
          System.out.println("Enter second number : ");
          b=Integer.parseInt(din.readLine());
                  System.out.println("Enter third number : ");
          c=Integer.parseInt(din.readLine());
          System.out.println("The Numbers are : "+a+" "+b+" "+c);
                  min=(a<b)?(a<c)?a:c:(b<c)?b:c;
                  System.out.println("Minimum = "+min);
       }
      catch(Exception e)
       {
          System.out.println("Error : "+e);
        }
      }
}
```

OUTPUT



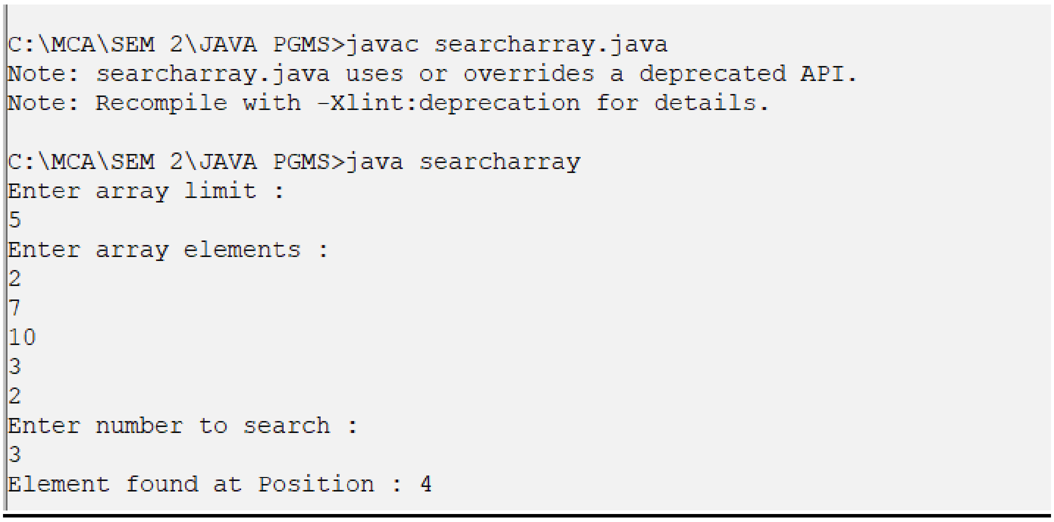
```
C:\MCA\SEM 2\JAVA PGMS>javac min3.java
Note: min3.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\JAVA PGMS>java min3
Enter first number :
70
Enter second number :
80
Enter third number :
65
The Numbers are : 70 80 65
Minimum = 65
```

3. **WAP to search for a given element in an array.**

SOURCE CODE

```
import java.io.*;
class searcharray
{
  public static void main(String args[])
   {
     DataInputStream din = new DataInputStream(System.in);
     int a[],n,i,s;
     try
      {
        System.out.println("Enter array limit : ");
        n=Integer.parseInt(din.readLine());
              a=new int[n];
        System.out.println("Enter array elements : ");
        for(i=0;i<n;i++)
              {
                        a[i]=Integer.parseInt(din.readLine());
              }
              System.out.println("Enter number to search : ");
              s=Integer.parseInt(din.readLine());
              for(i=0;i<n;i++)
              {
                        if(a[i]==s)
                        {
                                  System.out.println("Element found at Position :
"+(i+1));
                                  break;
                        }
              }
              if(i>=n)
              {
                        System.out.println("Element not found");
              }
      }
     catch(Exception e)
      {
        System.out.println("Error : "+e);
      }
   }
}
```

OUTPUT



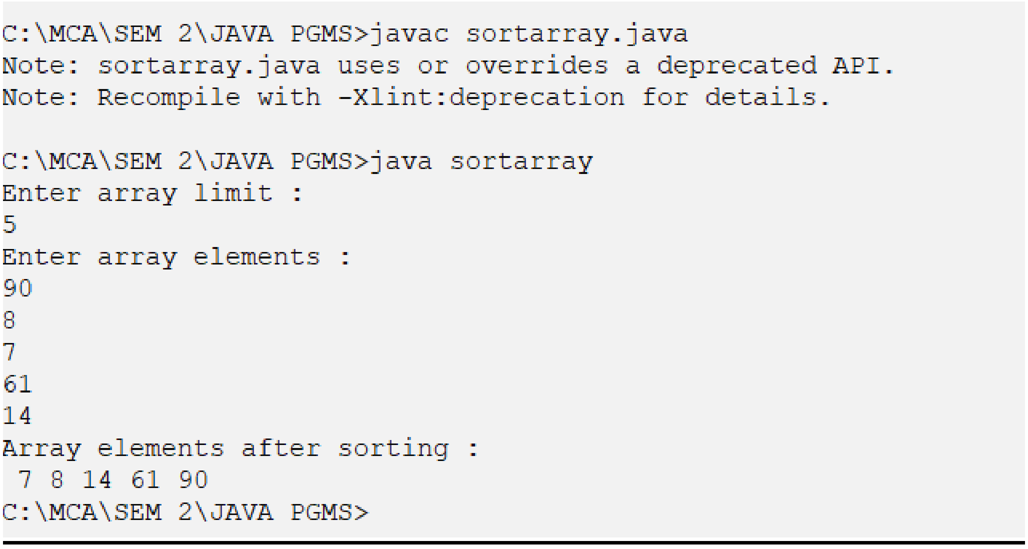
```
C:\MCA\SEM 2\JAVA PGMS>javac searcharray.java
Note: searcharray.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\JAVA PGMS>java searcharray
Enter array limit :
5
Enter array elements :
2
7
10
3
2
Enter number to search :
3
Element found at Position : 4
```

4. WAP to sort elements in an array in ascending order.

SOURCE CODE

```
import java.io.*;
class sortarray
{
  public static void main(String args[])
   {
     DataInputStream din = new DataInputStream(System.in);
     int a[],n,i,j,temp;
     try
      {
        System.out.println("Enter array limit : ");
        n=Integer.parseInt(din.readLine());
                    a=new int[n];
        System.out.println("Enter array elements : ");
        for(i=0;i<n;i++)
                {
                        a[i]=Integer.parseInt(din.readLine());
                }
                for(i=0;i<n;i++)
                {
                        for(j=0;j<n-1;j++)
                        {
                                if(a[j]>a[j+1])
                                {
                                        temp=a[j];
                                        a[j]=a[j+1];
                                        a[j+1]=temp;
                                }
                        }
                }
                System.out.println("Array elements after sorting : ");
                for(i=0;i<n;i++)
                        System.out.print(" "+a[i]);

      }
      catch(Exception e)
      {
        System.out.println("Error : "+e);
      }
    }
}
```

<u>OUTPUT</u>



```
C:\MCA\SEM 2\JAVA PGMS>javac sortarray.java
Note: sortarray.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\JAVA PGMS>java sortarray
Enter array limit :
5
Enter array elements :
90
8
7
61
14
Array elements after sorting :
 7 8 14 61 90
C:\MCA\SEM 2\JAVA PGMS>
```

5. Write a program to print the row wise and column wise sum of a 2D array.

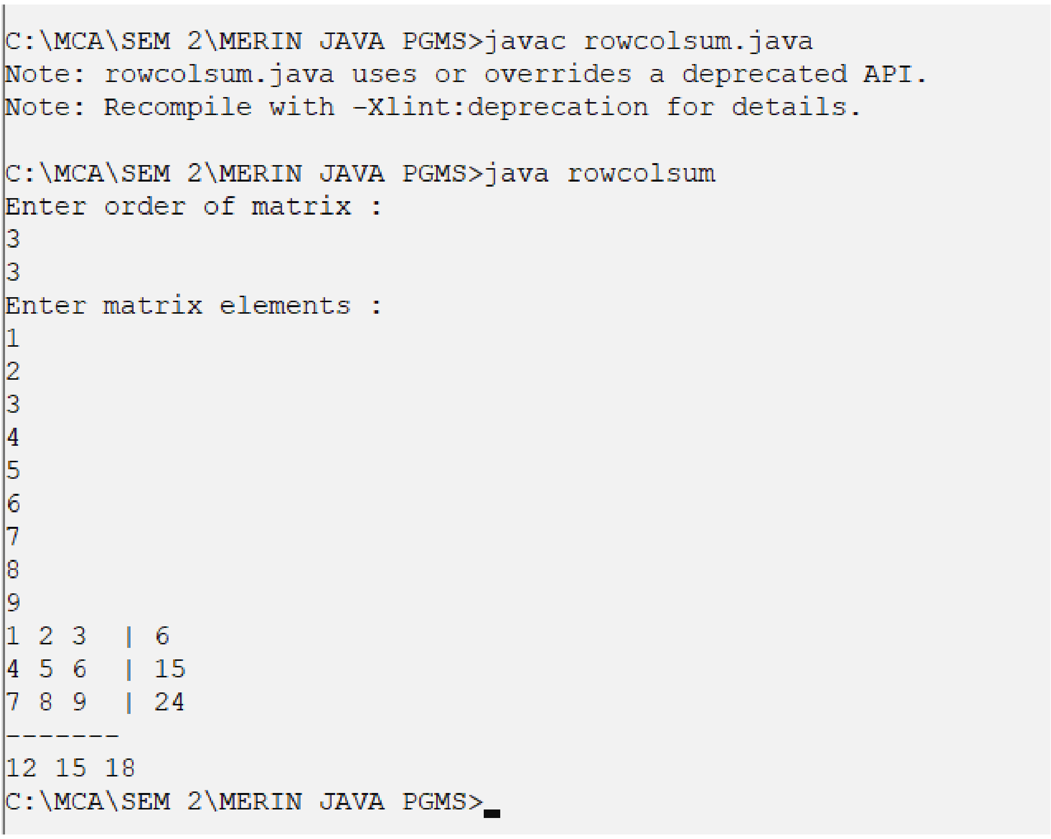
```
1 2 3 | 6
2 1 1 | 4
.  .  .

3 3 4
```

SOURCE CODE

```
import java.io.*;
class rowcolsum
{
  public static void main(String args[])
   {
     DataInputStream din = new DataInputStream(System.in);
     int a[][],n,m,i,j,row,col[];
     try
      {
        System.out.println("Enter order of matrix : ");
        n=Integer.parseInt(din.readLine());
                    m=Integer.parseInt(din.readLine());
                    a=new int[n][m];
                    col=new int[m];
        System.out.println("Enter matrix elements : ");
        for(i=0;i<n;i++)
                    {
                            for(j=0;j<m;j++)
                            {
                                    a[i][j]=Integer.parseInt(din.readLine());
                                    col[j]=0;
                            }

                    }
                    for(i=0;i<n;i++)
                    {
                            row=0;
                            for(j=0;j<m;j++)
                            {
                                    System.out.print(a[i][j]+" ");
                                    row=row+a[i][j];
                                    col[j]=col[j]+a[i][j];
                            }
                            System.out.println(" | " + row);

                    }
                    System.out.println("-------");
                    for(i=0;i<m;i++)
                    {
                            System.out.print(col[i]+" ");
```

```
                  }

      }
     catch(Exception e)
      {
        System.out.println("Error : "+e);
      }
    }
}
```

OUTPUT



```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac rowcolsum.java
Note: rowcolsum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java rowcolsum
Enter order of matrix :
3
3
Enter matrix elements :
1
2
3
4
5
6
7
8
9
1 2 3  | 6
4 5 6  | 15
7 8 9  | 24
-------
12 15 18
C:\MCA\SEM 2\MERIN JAVA PGMS>
```

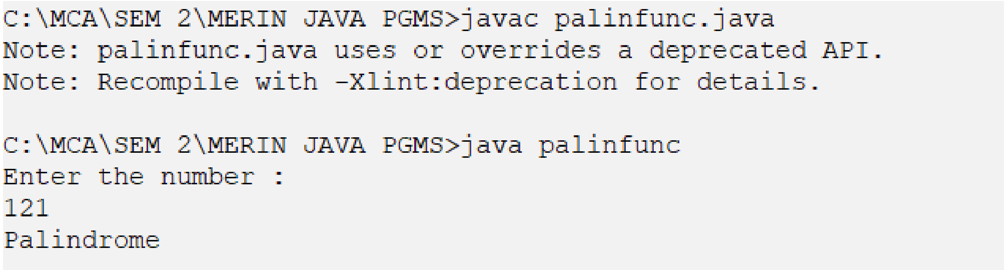
SECTION 3

1. **WAP with two functions to check for an integer palindrome. (Function1 should reverse the integer. Function2 should return 1,if it is a palindrome or else 0.)**

```
import java.io.*;
class palinfunc
{
  public static int reverse(int n)
  {
```

```
                int i,mod,rev=0;
                for(i=n;i!=0;i=i/10)
                {
                         mod=i%10;
                        rev=rev*10+mod;
                }
                return rev;
        }
        public static int checkp(int n)
        {
                if(reverse(n)==n)
                        return 1;
                else
                        return 0;
        }

        public static void main(String args[])
         {
          DataInputStream din = new DataInputStream(System.in);
          int n,c;
          try
           {
             System.out.println("Enter the number : ");
             n=Integer.parseInt(din.readLine());
                        c=checkp(n);
                        if(c==1)
                                System.out.println("Palindrome");
                        else
                                System.out.println("Not palindrome");

           }
          catch(Exception e)
           {
             System.out.println("Error : "+e);
           }
         }
      }
```

OUTPUT



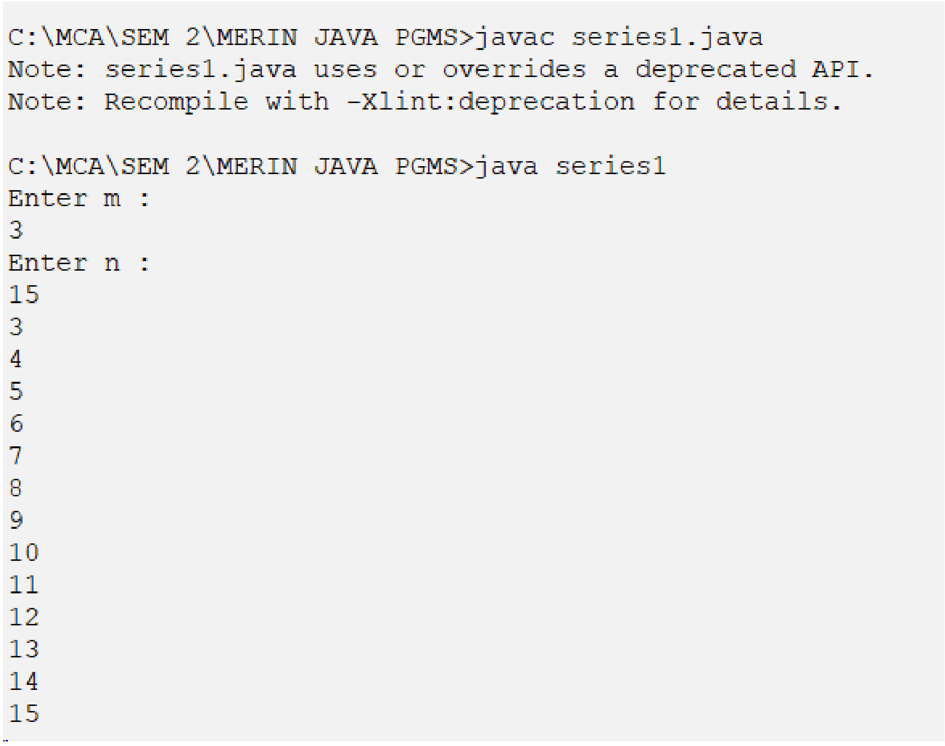
```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac palinfunc.java
Note: palinfunc.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java palinfunc
Enter the number :
121
Palindrome
```

2. WAP to display numbers from m to n using single while loop.

SOURCE CODE

```
//printing elements from m to n using single while loop
import java.io.*;
public class series1
{
  public static void main(String args[])
    {
      DataInputStream din = new DataInputStream(System.in);
      int m,n;
      try
       {
          System.out.println("Enter m : ");
          m=Integer.parseInt(din.readLine());
          System.out.println("Enter n : ");
          n=Integer.parseInt(din.readLine());
                while(m<=n)
                {
                        System.out.println(m++);
                }
       }
       catch(Exception e)
       {
          System.out.println("Error : "+e);
       }
    }
}
```

OUTPUT



```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac series1.java
Note: series1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java series1
Enter m :
3
Enter n :
15
3
4
5
6
7
8
9
10
11
12
13
14
15
```

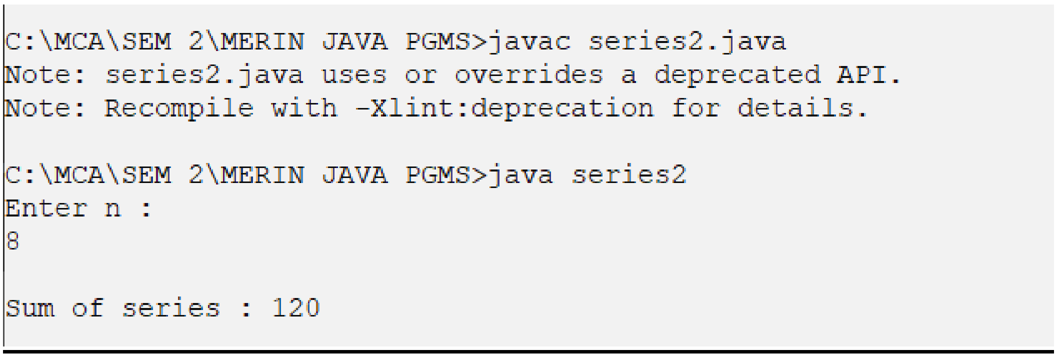
3. WAP to find the sum of the series 1+(1+2)+(1+2+3)+........... +(1+2+3+…+n) using a single while loop.

SOURCE CODE

```
//1+(1+2)+(1+2+3)+........... +(1+2+3+…+n) using a single while loop.

import java.io.*;
public class series2
{
  public static void main(String args[])
    {
      DataInputStream din = new DataInputStream(System.in);
      int n,i,s,sum;
      try
       {
          System.out.println("Enter n : ");
          n=Integer.parseInt(din.readLine());
                i=1;s=0;sum=0;
                while(i<=n)
                {
                        s=s+i;
                        sum=sum+s;
                        i++;
                }
                System.out.println("\nSum of series : "+ sum);
       }
       catch(Exception e)
```

```
          {
            System.out.println("Error : "+e);
          }
        }
      }
```

OUTPUT



```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac series2.java
Note: series2.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java series2
Enter n :
8

Sum of series : 120
```

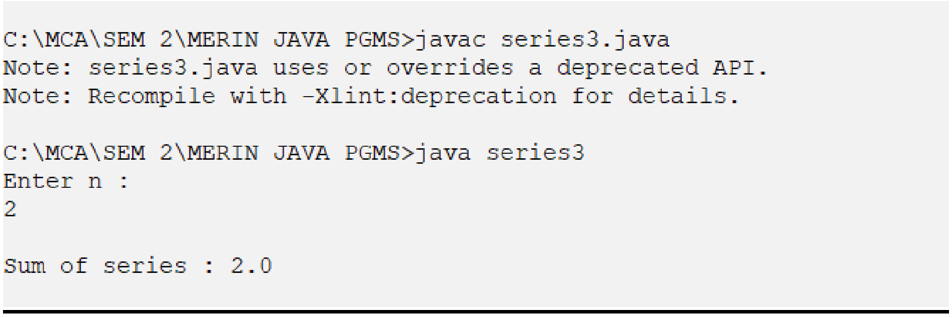
4. WAP to find the sum of 1+2/2!+3/3!+4/4!++n/n! using a single for loop.

SOURCE CODE

```
//sum of 1+2/2!+3/3!+4/4!++n/n! using a single for loop
import java.io.*;
public class series3
{
 public static int fact(int n)
 {
		int i,f=1;
		for(i=1;i<=n;i++)
			f=f*i;
		return f;
 }
 public static void main(String args[])
  {
    DataInputStream din = new DataInputStream(System.in);
    int n,i;
		float sum;
    try
     {
       System.out.println("Enter n : ");
       n=Integer.parseInt(din.readLine());
			sum=0;
			for(i=1;i<=n;i++)
			{
				sum=(float)sum+(i/fact(i));
			}
			System.out.println("\nSum of series : "+ sum);
```

```
                }
                catch(Exception e)
            {
              System.out.println("Error : "+e);
            }
                }
    }
```

OUTPUT



```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac series3.java
Note: series3.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java series3
Enter n :
2

Sum of series : 2.0
```

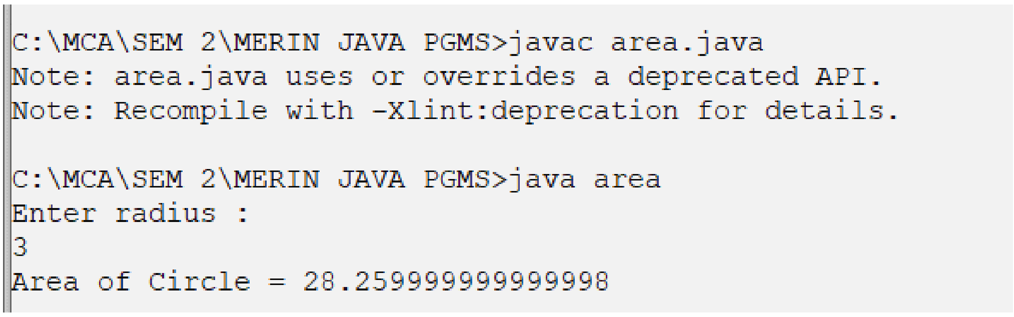
5. WAP to calculate area of a circle (functions with no argument and no return type.)

SOURCE CODE

```
import java.io.*;
public class area
{

  public static void areacal()throws IOException
  {
      int r;
      DataInputStream din = new DataInputStream(System.in);
      System.out.println("Enter radius : ");
      r=Integer.parseInt(din.readLine());
      System.out.println("Area of Circle = "+(3.14*r*r));

  }
  public static void main(String args[])
    {
     try
     {
      areacal();
     }
     catch(Exception e)
     {
```

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

OUTPUT



```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac area.java
Note: area.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java area
Enter radius :
3
Area of Circle = 28.259999999999998
```

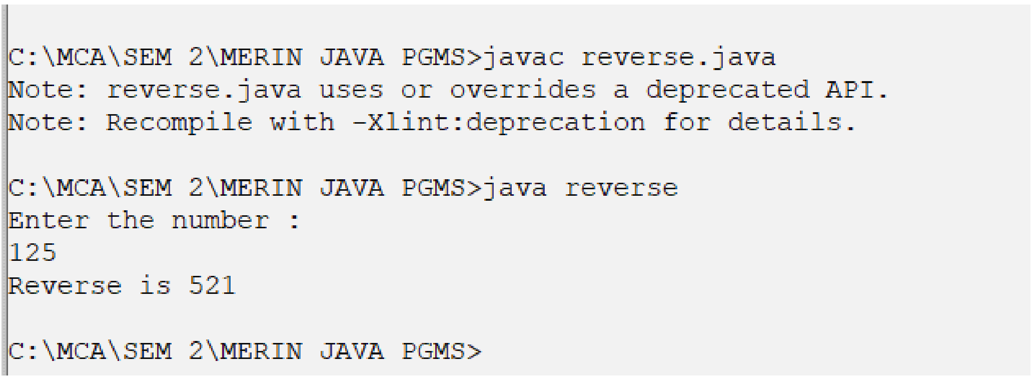
6. WAP to reverse a number (functions with argument and no return type.)

SOURCE CODE

```
import java.io.*;
class reverse
{
  public static int reverse(int n)
  {
     int i,mod,rev=0;
     for(i=n;i!=0;i=i/10)
     {
               mod=i%10;
              rev=rev*10+mod;
     }
     return rev;
  }

  public static void main(String args[])
   {
    DataInputStream din = new DataInputStream(System.in);
    int n,r;
    try
     {
       System.out.println("Enter the number : ");
       n=Integer.parseInt(din.readLine());
             r=reverse(n);
             System.out.println("Reverse is "+r);

     }
    catch(Exception e)
     {
       System.out.println("Error : "+e);
```

```
      }
    }
}
```

OUTPUT



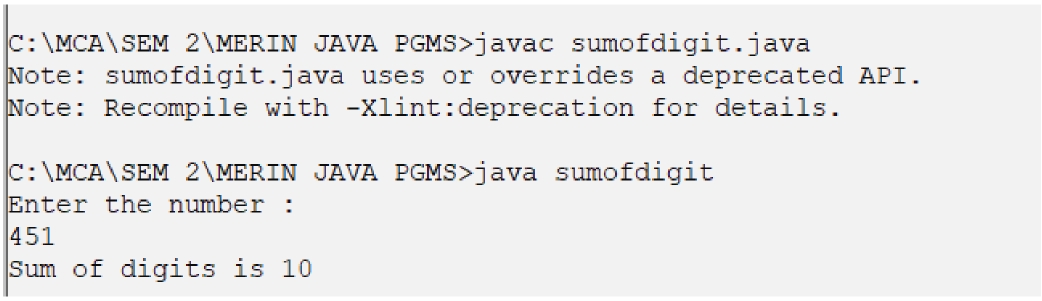
```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac reverse.java
Note: reverse.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java reverse
Enter the number :
125
Reverse is 521

C:\MCA\SEM 2\MERIN JAVA PGMS>
```

7. **WAP to calculate sum of digits of a number (functions with argument and return type.)**

SOURCE CODE

```
import java.io.*;
class sumofdigit
{
  public static int sumd(int n)
  {
     int i,mod,sum=0;
     for(i=n;i!=0;i=i/10)
     {
              mod=i%10;
             sum=sum+mod;
     }
     return sum;
  }

  public static void main(String args[])
   {
    DataInputStream din = new DataInputStream(System.in);
    int n,r;
    try
     {
       System.out.println("Enter the number : ");
       n=Integer.parseInt(din.readLine());
             r=sumd(n);
             System.out.println("Sum of digits is "+r);

     }
    catch(Exception e)
     {
       System.out.println("Error : "+e);
     }
   }
}
```

OUTPUT



```
C:\MCA\SEM 2\MERIN JAVA PGMS>javac sumofdigit.java
Note: sumofdigit.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\MERIN JAVA PGMS>java sumofdigit
Enter the number :
451
Sum of digits is 10
```

8. **WAP to calculate sum of n even numbers (functions with no argument and return type.)**

SECTION 4

1. **WAP with nested functions to find the maximum of three numbers. Functi on1 should take in two arguments and find the maximum. Function2 should take in the third number and the maximum from function1 to find the maximum.)**

2. **WAP to find the factorial of n, using recursion.**

3. **WAP to display numbers from n to 1 and vice versa, using recursion.**
4. **Using constructors, implement the operations of a queue.**
5. **Create a class complex having a real and imaginary part. Provide function s for read, display, add and multiplying two complex numbers.**
6. **WAP to display even numbers upto ‘n’ using a static function.**

SECTION 5

1. **WAP (menu driven) to demonstrate method overriding in java, by displaying details of a student, and a teacher.**
2. **Create a class for employee having eno,ename and esal as data members. P rovide functions for reading and displaying employee details. (Accept information of n employees in the main function, display th e same and search for an emp (using eno)).**

SECTION 6

1. **Program to implement run time polymorphism in Java using interface, wrt calculating area of a triangle.**
2. **Create an interface Shape having two prototypes disp() and calc(), to diplay the shape and calculate volume respectively. Create two classes circle and rectangle which implements the above interface. In the main function create a reference of Shape depending on the user-choice.**
3. **WAP to implement a function using call by value to swap two float numbers.**
4. **WAP to implement a function using call by reference to find the square root of a given number.**