

① HOLLOW BLOCK MATRIX:-

A Hollow Block Matrix is a matrix with elements only on the Border of Perimeter, and the interior elements are filled with zeros.

PROGRAM FOR HOLLOW BLOCK MATRIX:

```
import java.util.Scanner;

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

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

        int rows = scanner.nextInt();

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

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

int [][] matrix = new int[rows][columns];

for (int i=0; i<rows; i++)
{
    for (int j=0; j<columns; j++)
    {
        if (i==0 || i==rows-1 || j==0 ||
            j==columns-1)
        {
            matrix[i][j]=1;
        }
    }
}

for (int i=0; i<rows; i++)
{
    for (int j=0; j<columns; j++)
    {
        System.out.println [matrix[i][j]
                               + " "];
```

```
};

System.out.println ();

}

Scanner.close();

}
```

② FACTORIAL OF N PRIME NUMBERS

```
import java.util.Scanner;

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

        System.out.println("Enter the value
            of N: ");

        int n = scanner.nextInt();
        int count = 0;
        int number = 2;

        while (count < n)
        {
            if (isprime(number))
            {
                long factorial = calculateFactorial
                    (number);

                System.out.println ("factorial of
```

```
prime" + number + ":" + factorial);

                count++;
            }
            number++;
        }
        scanner.close();
    }

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

```
private static long calculate
    factorial (int num)
{
    long factorial = 1;
    for (int i = 2; i <= num; i++)
    {
        factorial *= i;
    }
    return factorial;
}
```