**Return the prime numbers that are strictly greater than m and strictly less than n, for a given an integer m and n. (in output each row must have 5 numbers only)**

|  |
| --- |
| **Example 1:**  Input: m = 2 and n = 10  Output: 3, 5, 7 |
| **Example 2:**  Input: m = 1000 and n = 1200  Output: 1009, 1013, 1019, 1021, 1031, 1033,  1039, 1049, 1051, 1061, 1063, 1069,  1087, 1091, 1093, 1097, 1103, 1109,  1117, 1123, 1129, 1151, 1153, 1163,  1171, 1181, 1187, 1193 |

**ANSWER**

 package prime;

class prime

 {

    public static void main(String[] args)

    {

        int m=2;

        int n=10;

        series(m, n);

    }

    public static void series(int m,int n)

    {

        for(int i=m;i<=n;i++)

        {

            if(primecheck(i))

            {

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

            }

        }

    }

    public static boolean primecheck(int n)

    {

        if(n<=1)

        {

            return false;

        }

        for(int i=2;i\*i<=n;i++)

        {

            if(n%i==0)

            {

                return false;

            }

        }

        return true;

    }

}

# \*\*\*\*\*\*\*\*\*\*\*\*

**Take 3 numbers as input and print the largest number (without using logical AND (&&))**

Example 1:

Input: 3 5 1

Output: 5

Explanation: 5 is greater than 3 and 1.

package largest;

public class largest

{

    public static void main(String[] args) {

       int a=3;

       int b=5;

       int c=1;

       int temp=0;

       if(a<b)

       {

        temp=a;

        a=b;

        b=temp;

       }

       if(b<c)

       {

        temp=b;

        b=c;

        c=temp;

       }

       System.out.println(a);

    }

}

# \*\*\*\*\*\*\*\*\*\*\*\*

**To find out whether the given String is Palindrome or not.**

|  |
| --- |
| **Example 1:**  Input: s = "amma"  Output: true  Explanation: "amma" is a palindrome. |
| **Example 2:**  Input: s = "hello"  Output: false  Explanation: "hello" is not a palindrome. |

 package palindrome;

public class palindrome {

    public static void main(String[] args) {

        String s = "malayalam";

        int i = 0;

        int j = s.length() - 1;

        boolean flag = true;

        while (i < j) {

            if (s.charAt(i) != s.charAt(j)) {

                flag = false;

                break;

            }

            i++;

            j--;

        }

        if (flag) {

            System.out.println("Palindrome");

        } else {

            System.out.println("Not a Palindrome");

        }

    }

}

# \*\*\*\*\*\*\*\*\*\*\*\*

**Return sum the factorial of a given number**

|  |
| --- |
| **Example 1:** Input: n = 4  Output: 7  Explanation: 7 = 1 + 2 + 4 |
| **Example 2:**  Input: n = 10  Output: 17  Explanation: 17 = 1 + 2 + 5 + 10 |

package factor;

public class factor

 {

    public static void main(String[] args)

     {

        int n=4;

        int total=0;

        for(int i=1;i<=n;i++)

        {

            if(n%i==0)

            {

                System.out.println(i);

                total+=i;

            }

        }

        System.out.println("total : "+total);

    }

}

# \*\*\*\*\*\*\*\*\*\*\*\*

Print the below pattern

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

package pattern;

public class pattern

{

    public static void main(String[] args)

     {

        int n=5;

        for(int i=1;i<=n;i++)

        {

            for(int j=1;j<=i;j++)

            {

                System.out.print(j+" ");

            }

            System.out.println("");

        }

    }

}

**\*\*\*\*\*\*\*\*\*\*\*\***