

- To check whether the number is **prime number**:

```
package programs_on_numbers;

public class PrimeNumber {

    public static void main(String[] args) {

        int number=11;

        int c=0;

        for (int i=2 ; i<=number ;i++) {

            if(number%i==0) {

                c++;

            }

        }

        if(c==1) {

            System.out.println(number + " : is a prime number");

        }

        else {

            System.out.println(number + " : is not a prime

number");

        }

    }

}
```

Output: 11 : is a prime number

- Print prime number that lies between 50 to 100 : **Prime Number between Range**

```
package programs_on_numbers;

public class PrimeNumberBetweenRange {

    public static void main(String[] args) {

        int number =100;

        int c=0;

        System.out.print("Prime numbers are :");

        for (int i=50 ; i<=number ;i++) {

            for (int j=2 ; j<=i ; j++) {

                if(i%j==0) {

                    c++;

                }

            }

            if (c==1) {

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

            }

            else {

                c=0;

            }

        }

    }

}
```

Output:Prime numbers are : 53 59 61 67 71 73 79 83 89 97

- To reverse the given number : **Reverse Number**

```
package programs_on_numbers;

public class ReverseNumber {

    public static void main(String[] args) {

        int number = 12345;

        int rev =0;

        int rem;

        int i=1;

        while (i<=number) {

            rem = number%10;

            rev = rev*10 + rem;

            number = number/10;

        }

        System.out.println("Reverse number is : " + rev);

    }

}
```

Output : Reverse number is : 54321

- Swap the given Numbers : **Swap number**

```
package numberProgram;

public class SwapNumber {

    public static void main(String[] args) {

        int a=12;

        int b=13;

        int c;

        c=a;

        a=b;

        b=c;

        System.out.println("The value of a is:"+a);

        System.out.println("The value of b is "+b);

    }

}
```

Output: The value of a is:13
 The value of b is:12

- To print whole number within a given range : **whole Number**

```
package programs_on_numbers;

public class WholeNumber {

    public static void main(String[] args) {

        int a=15;

        System.out.print("Whole numbers are : ");

        for(int b=0;b<=a;b++) {

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

        }

    }

}
```

Output: Whole numbers are : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

- To print Natural number within a given range : **Natural Number**

```
package programs_on_numbers;

public class NaturalNumbers {

    public static void main(String[] args) {

        System.out.print("Natural numbers are : ");

        int a=20;

        for (int b=1; b<=a; b++) {

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

        }

    }

}
```

Output: Natural numbers are : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20

- To print Odd number within a given range : **Odd Number**

```
package numberProgram;

public class Oddnumber {

    public static void main(String[] args) {

        int number = 20;

        System.out.print("Odd numbers are : ");

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

            if (i%2==0) {

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

            }

        }

    }

}
```

Output: Odd numbers are : 0 2 4 6 8 10 12 14 16 18 20

- To print ODD-EVEN numbers within a given range : **ODD
EVEN NUMBER**

```
package numberProgram;

public class OddevenNumber {

    public static void main(String[] args) {

        for (int a=1; a<=20; a++) {

            if (a%2==0) {

                System.out.println("This is even number "+ a);

            }

            else {

                System.out.println("This is odd number :"+ a);

            }

        }

    }

}
```

Output :

```
This is odd number :1
This is even number 2
This is odd number :3
This is even number 4
This is odd number :5
This is even number 6
This is odd number :7
This is even number 8
This is odd number :9
This is even number 10
This is odd number :11
This is even number 12
This is odd number :13
This is even number 14
This is odd number :15
This is even number 16
This is odd number :17
This is even number 18
This is odd number :19
This is even number 20
```


- To print Fibonacci series numbers within a given range : **Fibonacci series**

```
package numberProgram;

public class FibonnasiSeries {

    public static void main(String[] args) {

        int m=12;
        int a=0;
        int b=1;
        int c;

        int i=0;

        System.out.print("FibonnasiSeries is as :");

        while (i<=m) {

            c=a+b;

            a=b;
            b=c;
            i++;

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

        }

    }

}
```

Output: FibonnasiSeries is as : 1 2 3 5 8 13 21 34 55 89 144 233 377

- To check whether the given number is palindrome or not : **Palindrome Number**

```
package programs_on_numbers;

public class PalindromeNumber {

    public static void main(String[] args) {

        int number=121;
        int rem;
        int rev=0;
        int i=0;
        int pal = number;

        while (i<number) {

            rem=number%10;

            rev=rev*10+rem;

            number=number/10;
        }

        System.out.println(rev);

        if (rev == pal) {

            System.out.println(rev + " : is a palindrome
number");
        }

        else {

            System.out.println(rev + " : is not a palindrome
number");
        }

    }

}
```

Output: 121
121 : is a palindrome number

➤ To print factorial of given number : **Factorial**

```
package programs_on_numbers;

public class FactorialNumber {

    public static void main(String[] args) {

        int number=5;

        int fact=1;

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

            fact =fact*i;

        }

        System.out.println("Factorial of the "+ number + " is
:" + fact);
    }
}
```

Output: Factorial of the 5 is :120

- To check whether the given number is Armstrong number : **Armstrong number**

```
package programs_on_numbers;

public class AromstrongNumber {

    public static void main(String[] args) {

        int number = 1634 ;
        int rev =0 ;
        int rem ;
        int arms = number ;

        while (number > 0) {

            rem = number % 10 ;

            rev = rev + rem *rem * rem * rem;

            number = number / 10 ;

        }

        System.out.println(rev);

        if (rev ==arms) {

            System.out.println(rev + " is a armstrong
number");
        }
        else {

            System.out.println(rev+ " is not a armstrong
number");
        }
    }
}
```

Output : 1634
1634 is a armstrong number

➤ Print numbers 1_10 without using for loop :

```
package programs_on_numbers;

public class Print1_10WithoutUsingForLoop {

    public void myMethod(int a){

        if(a<=10){

            System.out.println(a);

            myMethod(a+1);

        }

    }

    public static void main(String[] args) {

        Print1_10WithoutUsingForLoop w = new
Print1_10WithoutUsingForLoop();

        w.myMethod(1);

    }

}
```

Output :

```
1
2
3
4
5
6
7
8
9
10
```

- Find Largest Number from given Number : **Largest Number**

```
package programs_on_numbers;

public class Largest_Number {

    public static void main(String[] args) {
        int a=56;
        int b=567;
        int c =36;
        int d= 389;

        if(a>b&& a>c&&a>d)
        {
            System.out.println(a+ " is a largest number");
        }

        else if(b>a&& b>c&&b>d)
        {
            System.out.println(b+ " is a largest number");
        }

        if(c>a&& c>b&&c>d)
        {
            System.out.println(c+ " is a largest number");
        }

        else if(d>a&& a>b&&a>c)
        {
            System.out.println(d+ " is a largest number");
        }

    }

}
```

Output :

567 is a largest number

- How to compare the numbers (find big one) without using if else condition:

```
package programs_on_numbers;

public class CompareNumberWithoutUsingIFCondition {

    public static void main(String args[])
    {
        int x=10;
        int y=20;

        int max = (x >y) ? x : y;
        System.out.println("The largest numbers is: "+max);
    }
}
```

Output: The largest numbers is: 20

- To print sum of all even numbers from 1 to 15 : **SUM OF EVEN NUMBERS**

```
package programs_on_numbers;

public class EvenNumberSum {

    public static void main(String args [ ])
    {
        int i,sum=0;
        for(i=1;i<=15;i++)
        {
            if(i%2==0)
            {
                sum=sum+i;
            }
        }

        System.out.println("Final sum value is: "+sum);
    }
}
```

Output: Final sum value is: 56

- To print product of all even numbers from 1 to 10: **PRODUCT OF EVEN NUMBERS**

```
package programs_on_numbers;

public class EvenNumbersProduct {

    public static void main(String args [ ])
    {
        int i,product=1;

        for(i=1;i<=10;i++)
        {
            if(i%2==0)
            {
                product=product*i;
            }

        }

        System.out.println("Final product value is: "+product);
    }
}
```

Output : Final product value is: 3840

- To swap two numbers without using third variable : **SWAP WITHOUT USING THIRD VARIABLE**

```
package programs_on_numbers;

public class SwapWithoutUsingThirdVariable {

    public static void main(String[] args) {

        int a=10;

        int b=20;

        a=a+b; //a=10+20---->a=30

        b=a-b; //b=30-20----->b=10

        a=a-b; //a=30-10----->a=20

        System.out.println("After swapping : " +a+ " " +b);

        System.out.println("Value of a is :"+ a);

        System.out.println("Value of a is :"+ b);

    }

}
```

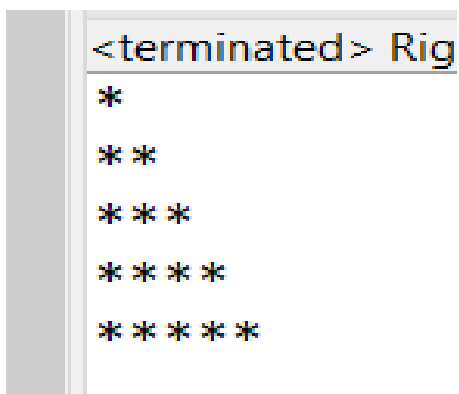
OUTPUT: After swapping : 20 10
 Value of a is :20
 Value of a is :10

Pattern Programs :

Pattern 1:

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

```
package patternProgram;  
  
public class RightAngle {  
    public static void main(String[] args) {  
        for (int a=1;a<=5;a++) {  
            for (int b=1; b<=a;b++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```



```
<terminated> Rig  
*  
**  
***  
****  
*****
```

Pattern 2:

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

```
package program_on_patterns;

public class RightAnglePattern2 {

    public static void main(String[] args) {

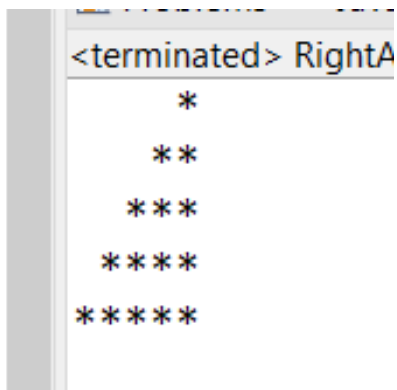
        for(int a=1;a<=5; a++) {

            for (int b=4; b>=a;b--) {

                System.out.print(" ");
            }
            for (int c=1; a>=c; c++) {

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

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



Pattern 3 :

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

```
package patternProgram;
```

```
public class Pyramid {
```

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

```
        for(int a=1;a<=5;a++) {
```

```
            for (int b=4; b>=a;b--) {
```

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

```
            }
```

```
            for (int c=1;a>=c; c++ ) {
```

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

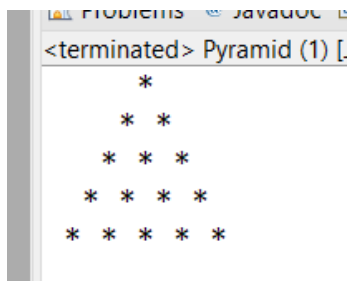
```
            }
```

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

```
        }
```

```
    }
```

```
}
```



Pattern 4:

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

```
public class Pyramid2 {

    public static void main(String[] args) {

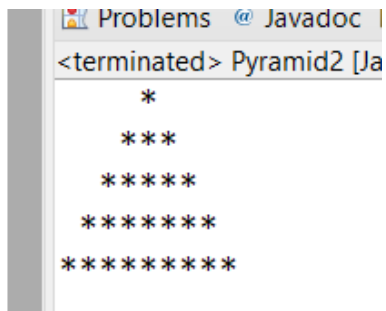
        for (int a=1;a<=5;a++)
        {
            for (int b=4; b>=a;b--)
            {

                System.out.print(" ");
            }

            for (int c=1; a>=c;c++)
            {
                System.out.print("*");
            }

            for (int d=2; a>=d; d++)
            {
                System.out.print("*");
            }

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



Pattern 5:

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

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

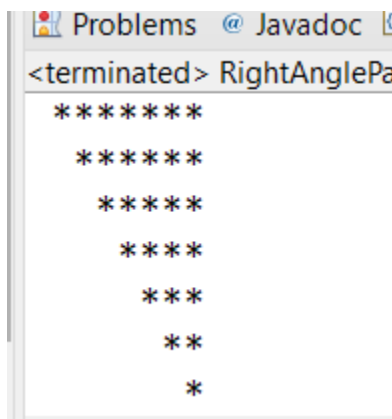
        for (int a=1; a<=7;a++) {

            for (int b=1; a>=b; b++)
            {
                System.out.print(" ");
            }
            for (int c=7; c>=a; c--)
            {
                System.out.print("*");
            }

            System.out.println();

        }

    }
}
```



Pattern 6:

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

```
public class HollowPattern {

    public static void main(String[] args) {

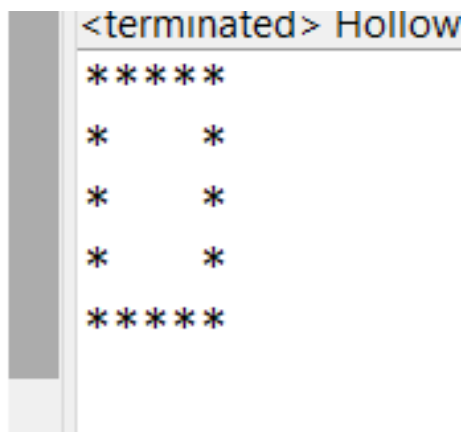
        for (int a=1; a<=5; a++) {

            for (int b=1;b<=5;b++) {

                if (a>=2 && a<=4 && b>=2 && b<=4) {

                    System.out.print(" ");
                }
                else {

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

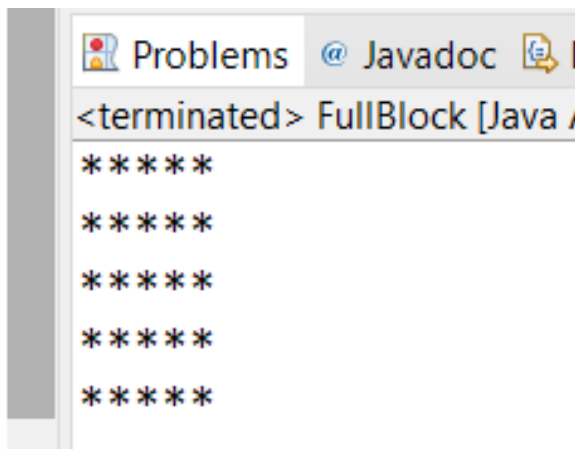


```
<terminated> Hollow
*****
*      *
*      *
*      *
*****
```


Pattern 7:

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

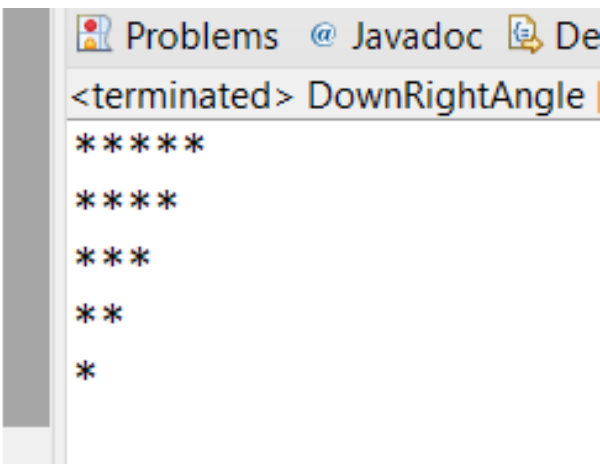
```
public class FullBlock {  
    public static void main(String[] args) {  
        for (int a=1;a<=5;a++) {  
            for (int b=1 ; b<=5; b++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```



Pattern 8:

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

```
public class DownRightAngle {  
    public static void main(String[] args) {  
        for (int a=1;a<=5;a++) {  
            for(int b=5; b>=a; b--) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```



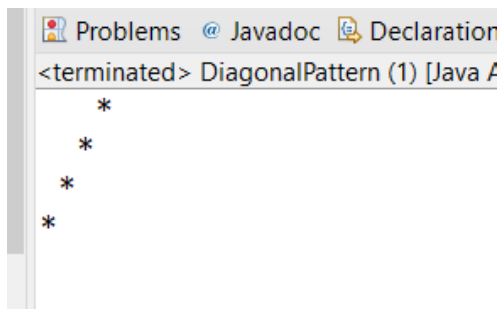
Pattern 9 :

```
    *
   *
  *
 *
```

```
public class DiagonalPattern {

    public static void main(String[] args) {

        for (int i=1;i<=4;i++)
        {
            for (int j=4;j>i;j--)
            {
                System.out.print(" ");
            }
            for (int k=1; k<=i; k++)
            {
                if (i>=2 && k>1)
                {
                    System.out.print(" ");
                }
                else
                {
                    System.out.print("*");
                }
            }
            System.out.println();
        }
    }
}
```



Pattern 10:

```
*
 *
  *
   *
    *
```

```
public class DiagonalPattern2 {

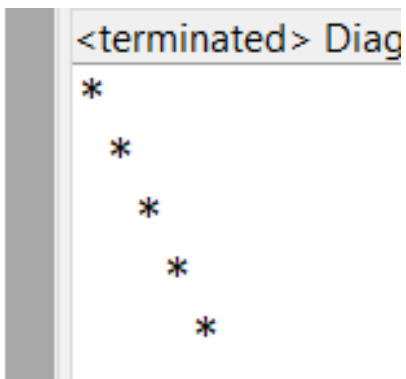
    public static void main(String[] args) {

        for (int a=1; a<=5; a++)
        {

            for (int b=1; a>=b; b++) {

                if (a>=2 && b<=(a-1)) {

                    System.out.print(" ");
                }
                else {
                    System.out.print("*");
                }
            }
            System.out.println();
        }
    }
}
```



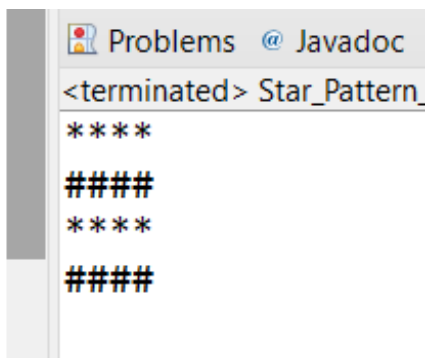
```
<terminated> Diag
*
 *
  *
   *
    *
```

Pattern 11 :

####

####

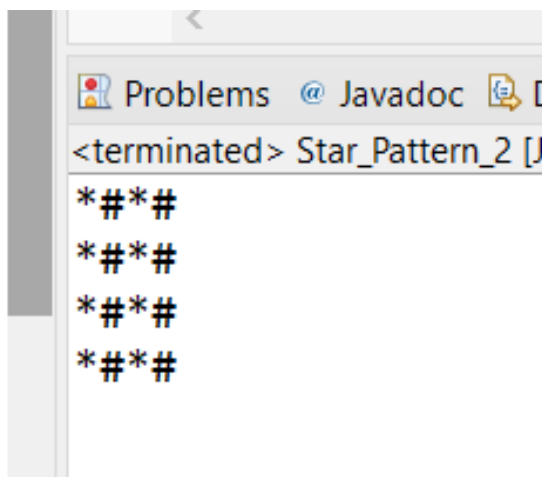
```
public class Star_Pattern_1 {  
    public static void main(String args[ ])  
    {  
        for(int i=1;i<=4;i++)  
        {  
            for(int j=1;j<=4;j++)  
            {  
                if(i%2==0) //(i==2 || i==4)  
                {  
                    System.out.print("#");  
                }  
                else  
                {  
                    System.out.print("*");  
                }  
            }  
            System.out.println(" ");  
        }  
    }  
}
```



Pattern 12 :

```
*##  
*##  
*##  
*##
```

```
public class Star_Pattern_2 {  
  
    public static void main(String args[ ])  
    {  
        for(int i=1;i<=4;i++)  
        {  
            for(int j=1;j<=4;j++)  
            {  
                if(j==2||j==4)/(j%2==0)  
                {  
                    System.out.print("#");  
                }  
                else  
                {  
                    System.out.print("*");  
                }  
            }  
            System.out.println(" ");  
        }  
    }  
}
```



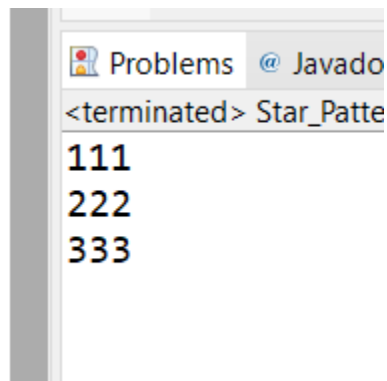
Pattern 13:

111

222

333

```
public class Star_Pattern_3 {  
    public static void main(String args[ ])   
    {  
        for(int i=1;i<=3;i++)  
        {  
            for(int j=1;j<=3;j++)  
            {  
                System.out.print(i);  
            }  
            System.out.println();  
        }  
    }  
}
```



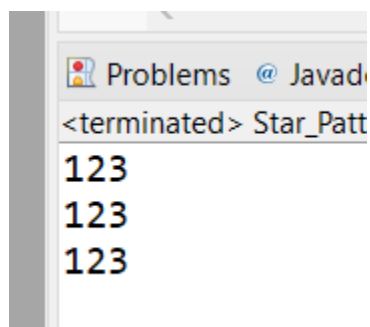
Pattern 14:

123

123

123

```
public class Star_Pattern_4 {  
    public static void main(String args[ ])   
    {  
        for(int i=1;i<=3;i++)  
        {  
            for(int j=1;j<=3;j++)  
            {  
                System.out.print(j);  
            }  
            System.out.println();  
        }  
    }  
}
```



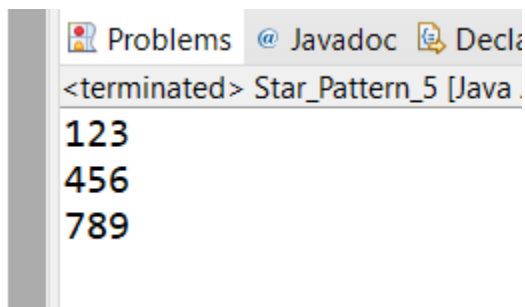
Pattern 15:

123

456

789

```
public class Star_Pattern_5 {  
  
    public static void main(String args[ ])   
    {  
        int a=1;  
        for(int i=1;i<=3;i++)  
        {  
            for(int j=1;j<=3;j++)  
            {  
                System.out.print(a);  
                a++;  
            }  
            System.out.println();  
        }  
    }  
}
```



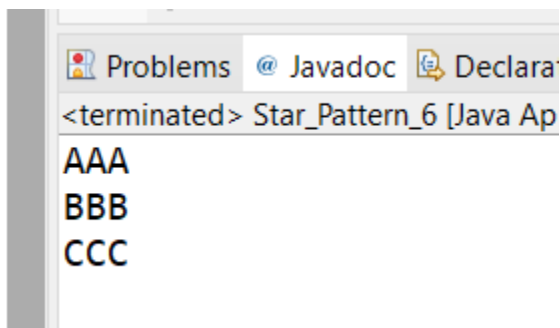
Pattern 16 :

AAA

BBB

CCC

```
public class Star_Pattern_6 {  
    public static void main(String args[ ])   
    {  
        char ch='A';  
        for(int i=1;i<=3;i++)  
        {  
            for(int j=1;j<=3;j++)  
            {  
                System.out.print(ch);  
            }  
            ch++;  
            System.out.println();  
        }  
    }  
}
```



Pattern 17:

ABC
ABC
ABC

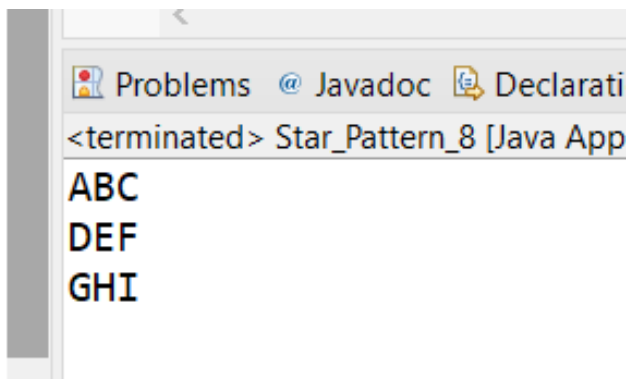
```
public class Star_Pattern_7 {  
    public static void main(String args[ ])   
    {  
        for(int i=1;i<=3;i++)  
        {  
            char ch='A';  
            for(int j=1;j<=3;j++)  
            {  
                System.out.print(ch);  
                ch++;  
            }  
            System.out.println();  
        }  
    }  
}
```



Pattern 18:

ABC
DEF
GHI

```
public class Star_Pattern_8 {  
    public static void main(String args[ ])   
    {  
        char ch='A';  
        for(int i=1;i<=3;i++)  
        {  
            for(int j=1;j<=3;j++)  
            {  
                System.out.print(ch);  
                ch++;  
            }  
            System.out.println();  
        }  
    }  
}
```

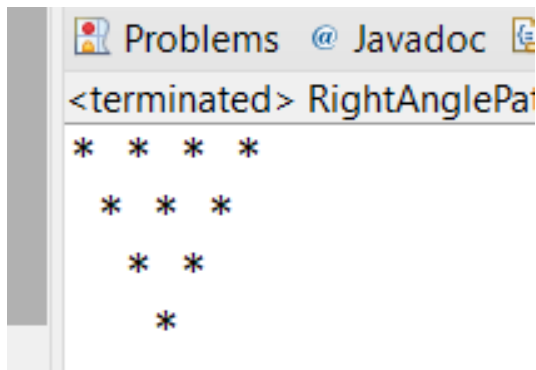


Pattern 18 :

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

```
public class Pyramid3 {

    public static void main(String args[ ])
    {
        for(int i=1;i<=4;i++)
        {
            for(int j=1;j<=4;j++)
            {
                if(i<=j)
                {
                    System.out.print("* ");
                }
                else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```

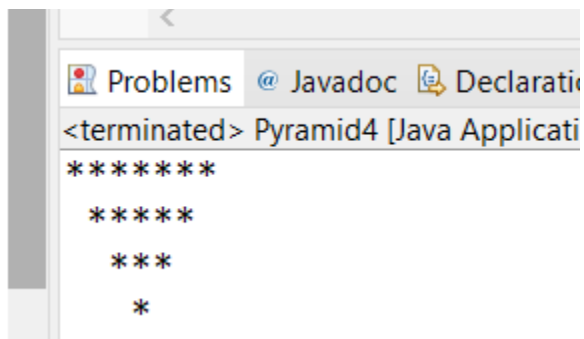


Pattern 20:

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

```
public class Pyramid4 {

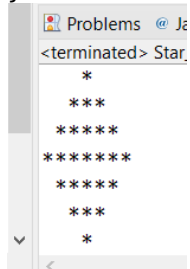
    public static void main(String args[])
    {
        int star=7,space=0;//(spaces before star)
        for(int i=1;i<=4;i++)
        {
            for(int j=1;j<=space;j++)
            {
                System.out.print(" ");
            }
            for(int k=1;k<=star;k++)
            {
                System.out.print("*");
            }
            star=star-2;
            space=space+1;
            System.out.println(" ");
        }
    }
}
```



Pattern 21 :

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

```
public class Star_Pattern_9 {
    public static void main(String args[])
    {
        int star=1,space=3;
        for(int i=1;i<=7;i++)
        {
            for(int j=1;j<=space;j++)
            {
                System.out.print(" ");
            }
            for(int k=1;k<=star;k++)
            {
                System.out.print("*");
            }
            if(i<=3 )
            {
                star=star+2;
                space=space-1;
            }
            else {
                star=star-2;
                space=space+1;
            }
            System.out.println(" ");
        }
    }
}
```

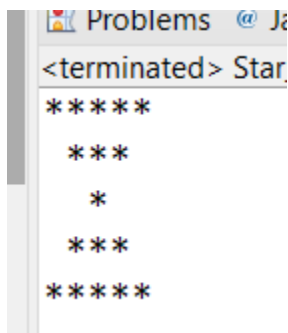


Pattern 22:

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

```
public class Star_Pattern_10 {

    public static void main(String args[])
    {
        int star=5,space=0;
        for(int i=1;i<=5;i++)
        {
            for(int j=1;j<=space;j++)
            {
                System.out.print(" ");
            }
            for(int k=1;k<=star;k++)
            {System.out.print("*");
            }
            if(i<=2 )
            {
                star=star-2;
                space=space+1;
            }
            else {
                star=star+2;
                space=space-1;
            }
            System.out.println(" ");
        }
    }
}
```

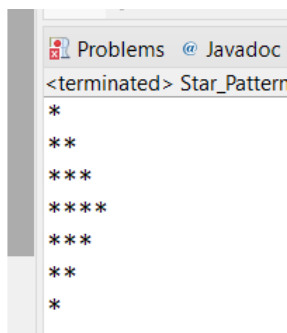


Pattern 23:

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

```
public class Star_Pattern_11 {

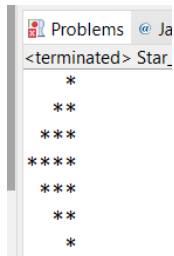
    public static void main(String args[])
    {
        int star=1,space=0;
        for(int i=1;i<=7;i++)
        {
            for(int k=1;k<=star;k++)
            {
                System.out.print("*");
            }
            if(i<=3 )
            {
                star=star+1;
            }
            else {
                star=star-1;
            }
            System.out.println(" ");
        }
    }
}
```



Pattern 24:

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

```
public class Star_Pattern_12 {
    public static void main(String args[])
    {
        int star=1,space=3;
        for(int i=1;i<=7;i++)
        {
            for(int j=1;j<=space;j++)
            {
                System.out.print(" ");
            }
            for(int k=1;k<=star;k++)
            {
                System.out.print("*");
            }
            if(i<=3 )
            {
                star=star+1;
                space=space-1;
            }
            else {
                star=star-1;
                space=space+1;
            }
            System.out.println(" ");
        }
    }
}
```



ARRAY PROGRAMS:

- To find Array Frequency of **Odd-Even Number**:

```
public class Array_Frequency_Odd_Even {  
  
    public static void main(String[] args) {  
  
        int a[]= {1,2,5,6,8,9};  
        int even=0;  
        int odd=0;  
        for(int i=0; i<a.length; i++)  
        {  
            if(a[i]%2==0)  
            {  
                even++;  
            }  
            else  
            {  
                odd++;  
            }  
        }  
        System.out.println("Frequency of even no : " +even);  
        System.out.println("Frequency of odd no : "+odd);  
    }  
}
```

Output :

```
Frequency of even no : 3  
Frequency of odd no : 3
```

➤ To find sum of Array Elements :**SUM**

```
public class SUM_OF_ARRAY_ELEMENTS {  
    public static void main(String[] args) {  
        int a[] = {10,15,7,20,55,87,18,47};  
        int sum=0;  
        for(int i=0; i<a.length; i++)  
        {  
            sum= sum + a[i];  
        }  
        System.out.println("Sum of all Array Elements is : " +sum);  
    }  
}
```

OUTPUT :

Sum of all Array Elements is : 259

➤ To find AVG of Array Elements :**AVG**

```
public class AVG_ARRAY_ELEMENTS {  
    public static void main(String[] args) {  
        int a[]= {10,26,29,34,76,49,53};  
        int sum=0;  
        for(int i=0; i<a.length; i++)  
        {  
            sum= sum+a[i];  
        }  
        System.out.println("The average of Array Elements is  : "  
+sum/a.length);  
    }  
}
```

OUTPUT :

The average of Array Elements is : 39

- To find BIG ELEMENT of Array Elements : **BIG ELEMENT**

```
public class BIG_ELEMENT_IN_ARRAY {  
  
    public static void main(String[] args) {  
        int a[] = {12,47,56,18,7,19,27};  
  
        int big = a[0];  
  
        for(int i=0; i<a.length; i++)  
        {  
            if(big<a[i])  
            {  
                big=a[i];  
            }  
        }  
  
        System.out.println("Biggest Element in Array is : " +big);  
    }  
}
```

OUTPUT :

Biggest Element in Array is : 56

- To find SMALLEST ELEMENT of Array Elements : **SMALLEST ELEMENT**

```
public class SMALL_ELEMENT_IN_ARRAY {  
  
    public static void main(String[] args) {  
        int a[] = {10,37,45,7,59,93};  
  
        int small = a[0];  
  
        for(int i=0; i<a.length; i++)  
        {  
            if(small>a[i])  
            {  
                small= a[i];  
            }  
        }  
  
        System.out.println("Smallest Element in Array is : "  
+small);  
    }  
}
```

OUTPUT :

Smallest Element in Array is : 7

➤ To find DUPLICATE ELEMENT of Array Elements : **DUPLICATE ELEMENT**

```
public class DUPLICATE_ELEMENTS_IN_ARRAY {  
  
    public static void main(String[] args) {  
        int a[] = {1, 25, 3, 1, 1, 2, 3, 3};  
  
        for(int i=0; i<a.length; i++)  
        {  
            for(int j=i+1; j<a.length; j++)  
            {  
                if(a[i]==a[j])  
                {  
                    System.out.println("Duplicates of Array is  
: " + a[j]);  
                }  
            }  
        }  
    }  
}
```

OUTPUT :

```
Duplicates of Array is : 1  
Duplicates of Array is : 1  
Duplicates of Array is : 3  
Duplicates of Array is : 3  
Duplicates of Array is : 1  
Duplicates of Array is : 3
```


- To find Frequency of Number in Array: **Frequency of Number in Array**

```
public class FREQUENCY_OF_NUMBER_IN_ARRAY {  
    public static void main(String[] args) {  
        int a[] = {10,15,12,17,12,12,18,12};  
        int num=12;  
        int count=0;  
        for(int i=0; i<a.length; i++)  
        {  
            if(a[i]==num)  
            {  
                count++;  
            }  
        }  
        System.out.println("Frequency of " +num + " in array is : "  
+count);  
    }  
}
```

OUTPUT :

Frequency of 12 in array is : 4

- To find MISSING ELEMENT of Array Elements : **Missing Element in Array**

```
public class MISSING_ELEMENT_IN_ARRAY {  
    public static void main(String[] args) {  
        int a[] = {1,2,3,4,5,7,8,9,10};  
        int val=1;  
        for(int i=0; i<a.length; i++)  
        {  
            if(a[i]!=val)  
            {  
                break;  
            }  
            val++;  
        }  
        System.out.println("Missing Element in Array is : "+val);  
    }  
}
```

OUTPUT:

Missing Element in Array is : 6

- To find Positive And Negative Element Count of Array Elements : **Positive And Negative Element Count**

```
public class POSITIVE_AND_NEGATIVE_ELEMENT_COUNT_ARRAY {  
  
    public static void main(String[] args) {  
        int a[]= {-10,-20,15,48,-15,47,78,-45};  
  
        int possitiveCount=0;  
  
        int negativeCount=0;  
  
        for(int i=0; i<a.length; i++)  
        {  
            if(a[i]>0)  
            {  
                possitiveCount++;  
            }  
            else  
            {  
                negativeCount++;  
            }  
        }  
        System.out.println("Possitive count of Element is : "  
+possitiveCount);  
        System.out.println("Negative count of Element is : "  
+negativeCount);  
    }  
}
```

OUTPUT :

```
Possitive count of Element is : 4  
Negative count of Element is : 4
```

➤ To reverse the given array: **Reverse Array**

```
public class REVERSE_ARRAY {  
    public static void main(String[] args) {  
        int a[] = {1,20,25,14,23,78,45,12};  
        for(int i=a.length-1; i>=0; i--)  
        {  
            System.out.print(a[i]+ " ");  
        }  
    }  
}
```

OUTPUT :

12 45 78 23 14 25 20 1

- To find second highest element from an array without sorting: **SECOND HIGHEST ELEMENT**

```
public class SECOND_HIGHEST_ELEMENT_IN_ARRAY {  
    public static void main(String[] args) {  
        int a[]= {22,5,6,88,9};  
        int max=a[0];  
        int secmax=a[0];  
        for(int i=1;i<a.length;i++)  
        {  
            if(a[i]>max)  
            {  
                secmax=max;  
                max=a[i];  
            }  
            else if(a[i]>secmax)  
            {  
                secmax=a[i];  
            }  
        }  
        System.out.println("maximum value : "+max);  
        System.out.println("second maximum value : "+secmax);  
    }  
}
```

OUTPUT :

```
maximum value : 88  
second maximum value : 22
```

- To find second smallest element from an array without sorting: **SECOND SMALLEST ELEMENT**

```
public class SECOND_SMALLEST_ELEMENT_IN_ARRAY {  
  
    public static void main(String[] args) {  
  
        int a[] = {22,5,6,88,9};  
  
        int min=a[0],secmin=a[0];  
  
        for(int i=1;i<a.length;i++)  
        {  
            if(a[i]<min)  
            {  
                secmin=min;  
                min=a[i];  
            }  
            else if(a[i]<secmin)  
            {  
                secmin=a[i];  
            }  
        }  
  
        System.out.println("minimum value : "+min);  
  
        System.out.println("second minimum value : "+secmin);  
    }  
}
```

OUTPUT :

```
minimum value : 5  
second minimum value : 6
```

- To find all missing numbers from 1 to 60 from an array: **FIND ALL MISSING ELEMENTS IN ARRAY**

```
public class FIND_ALL_MISSING_ELEMENTS_IN_ARRAY {  
  
    public static void main(String[] args) {  
  
        boolean status=true;  
  
        int a[]= {22,17,4,46,8,2,56};  
  
        for(int j=1;j<=60;j++)  
        {  
            for(int i=0;i<a.length;i++) {  
  
                if(j==a[i])  
                {  
                    status=false;  
                    break;  
                }  
            }  
            if(status==true)//number is not present  
            {  
                System.out.print(" " +j);  
            }  
            status=true;//for every number status should be true  
        }  
    }  
}
```

OUTPUT :

```
1 3 5 6 7 9 10 11 12 13 14 15 16 18 19 20 21 23 24 25 26 27 28 29  
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 47 48 49 50 51 52  
53 54 55 57 58 59 60
```

- To Sort the array elements using for-each loop: **SORT_ARRAY**

```
public class SORT_ARRAY {  
  
    public static void main(String[] args) {  
        int [] jk = new int [5] ;  
        jk [0]=3;  
        jk [1]=8;  
        jk [2]=5;  
        jk [3]=11;  
        jk [4]=15;  
  
        System.out.println("*****Before Sorting of  
array*****");  
        int size = jk.length;  
        System.out.println(size);  
        for (int bb:jk) {  
  
            System.out.println(bb);  
        }  
  
        System.out.println("*****AfterSortingOfArray*****");  
        Arrays.sort(jk);  
        for (int cx:jk) {  
  
            System.out.println(cx);  
        }  
    }  
}
```

OUTPUT :

```
*****Before Sorting of array*****  
5  
3  
8  
5  
11  
15  
*****AfterSortingOfArray*****  
3  
5  
8  
11  
15
```


PROGRAMS ON STRING:

➤ To find **duplicate string** :

```
public class FIND_DUPLICATE_STRING {  
  
    public static void main(String[] args) {  
  
        String a[]= {"Rohit", "Rahul", "Rohit",  
"Rahul","Suryawanshi"};  
  
        for(int i=0; i<a.length; i++)  
        {  
            for(int j=i+1; j<a.length; j++)  
            {  
                if(a[i].equals(a[j]))  
                {  
                    System.out.println("Duplicate of String is  
: " +a[j]);  
                }  
            }  
        }  
    }  
}
```

OUTPUT :

```
Duplicate of String is : Rohit  
Duplicate of String is : Rahul
```

- To find frequency of alphabet in given string: **FREQUENCY OF ALPHABET**

```
public class FREQUENCY_OF_ALPHABET {  
    public static void main(String[] args) {  
        int counter =0;  
        String str = "I Love My Counnntry";  
        char c ='n';  
        for(int i=0; i<str.length(); i++)  
        {  
            if(c==str.charAt(i))  
            {  
                counter++;  
            }  
        }  
        System.out.println("Frequency of " +c+ " = " +counter);  
    }  
}
```

OUTPUT :

Frequency of n = 3

- Java program to check whether given string is palindrome string:
PALINDROME STRING

```
public class PALINDROME_STRING {  
  
    public static void main(String[] args) {  
  
        String rev="";  
  
        String str="MADAM";  
  
        String org= str;  
  
        for(int i=str.length()-1; i>=0; i--)  
        {  
            rev= rev+ str.charAt(i);  
        }  
        System.out.println(rev);  
  
        if(org.equals(rev))  
        {  
            System.out.println(rev+ " is a palindrome String");  
        }  
        else  
        {  
            System.out.println(rev+ " is not a palindrome  
String");  
        }  
    }  
}
```

OUTPUT :

MADAM
MADAM is a palindrome String

- Java program to reverse each word of given string : **REVERSE EACH WORD OF GIVEN STRING**

```
public class REVERSE_EACH_WORD_OF_GIVEN_STRING {  
    public static void main(String[] args) {  
        String a = "Hello I am Sourabh";  
        String word[] = a.split(" ");  
        for(String element: word)  
        {  
            System.out.print(" ");  
            for(int i=element.length()-1; i>=0; i--)  
            {  
                char b = element.charAt(i);  
                System.out.print(b);  
            }  
        }  
    }  
}
```

OUTPUT :

olleH I ma hbaruoS

- Java program to reverse each word of given string :
reverse string with spaces

```
public class REVERSE_EACH_WORD {  
  
    public static void main(String[] args) {  
  
        String a ="Rohit is a Automation tester";  
  
        String [] b= a.split(" ");  
  
        for(int i=b.length-1; i>=0; i--)  
        {  
            System.out.print(" ");  
  
            System.out.print(b[i]);  
  
        }  
  
    }  
  
}
```

OUTPUT :

tester Automation a is Rohit

- Java program to reverse the given string completely : **REVERSE STRING**

```
public class STRING_REVERSE_COMPLETE {  
    public static void main(String[] args) {  
        String rev = "";  
        String str = "I Love My Country";  
        for(int i=str.length()-1; i>=0; i--)  
        {  
            rev = rev + str.charAt(i);  
        }  
        System.out.println(rev);  
    }  
}
```

OUTPUT :

yrtnuoC yM evoL I

- To check count of e/E character present in a String s="javaEEdeve"

```
public class STRING_COUNT {  
    public static void main(String[] args) {  
        String s="javaEEdeve";  
        int count=0;  
        for(int i=0;i<s.length();i++)  
        {  
            if(s.charAt(i)=='e' || s.charAt(i)=='E')  
            {  
                count++;  
            }  
        }  
        System.out.println("Count of E/e is : "+count);  
    }  
}
```

OUTPUT :

Count of E/e is : 4

- To find smaller case vowels from string s="javadev" a.print vowels b.count vowels:**FIND COUNT OF VOWELS**

```
public class FIND_VOWELS_OF_GIVEN_STRING {  
    public static void main(String[] args) {  
        String s="javadev";  
        int count=0;  
        for(int i=0;i<s.length();i++)  
        {  
            if(s.charAt(i)=='a' || s.charAt(i)=='e' || s.charAt(i)=='i' ||  
                s.charAt(i)=='o' || s.charAt(i)=='u')  
            {  
                System.out.println(s.charAt(i));  
                count++;  
            }  
        }  
        System.out.println("Count of vowels in smaller case is :  
"+count);  
    }  
}
```

OUTPUT:

```
a  
a  
e  
Count of vowels in smaller case is : 3
```


- To count no of words present in string: **COUNT OF WORDS IN GIVEN STRING**

```
public class COUNT_OF_WORD_IN_STRING {  
    public static void main(String[] args) {  
        String s=" I am a java developer ";  
        System.out.println("Before trimming:"+s);  
        String s1=s.trim(); //helps to remove spaces from start and  
end //of sentence  
        System.out.println("After trimming:"+s1);  
        int count=1;  
        for(int i=0;i<s1.length();i++) {  
            if(s1.charAt(i)==' ' && s1.charAt(i+1)!=' ') {  
                count=count+1;  
            }  
        }  
        System.out.println("No of words are : "+count);  
    }  
}
```

OUTPUT :

```
Before trimming: I am a java developer  
After trimming:I am a java developer  
No of words are : 5
```

- To print all characters only once from string String s="javajavajavadevdevdev" :

PRINT_CHARACTERS_ONLY_ONCE_OF_STRING

```
public class PRINT_CHARACTERS_ONLY_ONCE_OF_STRING {  
  
    public static void main(String[] args) {  
  
        String s="javajavajavadevdev";  
  
        String un="";  
  
        for(int i=0;i<s.length();i++) {  
  
            char ch=s.charAt(i);  
  
            if(un.indexOf(ch)==-1) {  
                un=un+ch;  
            }  
  
        }  
  
        System.out.println("Unique string is :"+un);  
    }  
}
```

OUTPUT :

Unique string is :javde

- To count longest word from a string String s="I am a java developer":

COUNT OF LARGEST WORD

```
public class FIND_LARGEST_WORD_OF_GIVEN_STRING {  
  
    public static void main(String[] args) {  
  
        String s="I am a java developer";  
  
        String s1[]=s.split(" ");  
  
        System.out.println("length of array : "+s1.length);  
  
        for(int i=0;i<s1.length;i++)  
        {  
            System.out.print(s1[i]+"-");  
  
            System.out.println(s1[i].length());  
        }  
        int max=0;  
        for(int i=0;i<s1.length;i++)  
        {  
            if(s1[i].length()>max)  
            {  
                max=s1[i].length();  
            }  
        }  
  
        System.out.print("The longest word from the string : "+max);  
    }  
}
```

OUTPUT :

```
length of array : 5  
I-1  
am-2  
a-1  
java-4  
developer-9  
The longest word from the string : 9
```

- To calculate frequency of characters present in a string "javadev" :
FREQUENCY OF CHARACTERS IN GIVEN STRING

```
public class FREQUENCY_OF_CHARACTERS {  
    public static void main(String[] args) {  
        String str="javadev";  
        String s=str.toUpperCase();//s=JAVADEV  
        char[] s1=s.toCharArray();//{'J','A','V','A','D','E','V'}  
        for(char ch='A';ch<='Z';ch++)  
        {  
            int count=0;  
            for(int i=0;i<s1.length;i++)  
            {  
                if(ch==s1[i])  
                {  
                    count++;  
                }  
            }  
            if(count>0)  
                System.out.println(ch+"-"+count);  
        }  
    }  
}
```

OUTPUT :

A-2
D-1
E-1
J-1
V-2

➤ **STRING MODIFICATIONS:**

- ♦ WAP to replace e with a in given string "java development"
- ♦ WAP to replace "java" with "core java" in given "java development"
- ♦ WAP to remove spaces from given string "java development"
- ♦ WAP to remove all capital letters from string "jAvA DeVeloPer"
- ♦ WAP to remove all small letters from same string
- ♦ WAP to remove digits from string "ja123vaDEveloper"
- ♦ WAP to remove vowels from string "ja123vaDEveloper"

```
public class STRING_MODIFICATIONS {  
  
    public static void main(String[] args) {  
        String s="java development";  
        String r1=s.replace('e','a');  
        System.out.println(r1);  
        String r2=s.replaceAll("java","core java");  
        System.out.println(r2);  
        String r3=s.replaceAll(" ","");  
        System.out.println(r3);  
        String s1="jAvA DeVeloPer";  
        String r4=s1.replaceAll("[A-Z]","");  
        System.out.println(r4);  
        String r5=s1.replaceAll("[a-z]","");  
        System.out.println(r5);  
        String s2="ja123vaDEveloper";  
        String r6=s2.replaceAll("[0-9]","");  
        System.out.println(r6);  
        String r7=s2.replaceAll("[aeiouAEIOU]","");  
        System.out.println(r7);  
    }  
}
```

OUTPUT :

```
java davalopmant  
core java development  
javadevelopment  
jv eeloer  
AA DVP  
javaDEveloper  
j123vDvlpr
```

❖ Java Program based on user input (Scanner class):

➤ Factorial Number:

```
public class Factorial {  
    public static void main(String[] args) {  
        Scanner jk = new Scanner (System.in);  
        System.out.println("Enter your number?");  
        int number = jk.nextInt();  
        int i;  
        int fact=1;  
        for ( i=1; i<=number ;i++) {  
            fact = fact*i;  
        }  
        System.out.println("Your value is:"+fact);  
        jk.close();  
    }  
}
```

OUTPUT :

Enter your number?

5

Your value is:120

➤ **Palindrome Number:**

```
import java.util.Scanner;

public class Palindrome_Number {

    public void method2() {

        Scanner as =new Scanner (System.in) ;
        System.out.println("Enter your palindrome number?");
        int number = as.nextInt();
        int remain=0;

        while (number > 0) {
            remain = remain*10+ number%10;
            number = number/10;
        }
        System.out.println("your palindrome number is : "+remain);

        as.close();
    }
    public static void main(String[] args) {
        Palindrome_Number jk = new Palindrome_Number ();
        jk.method2();
    }
}
```

OUTPUT :

Enter your palindrome number?

121

your palindrome number is : 121

➤ **Reverse Number:**

```
import java.util.Scanner;

public class Reverse_number {

    public static void main(String[] args) {

        Scanner jk = new Scanner (System.in);

        System.out.println("Enter your number");

        int number =jk.nextInt();

        int rev =0;
        while (number >0) {

            rev = rev *10+ number %10;
            number = number/10;
        }
        System.out.println("Your reverse number is : " + rev);
        jk.close();
    }
}
```

OUTPUT :

Enter your number

12345

Your reverse number is : 54321

➤ Prime Number :

```
public class PRIME_NUMBER {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.println("Enter your number ");  
        int number= input.nextInt();  
        int c=0;  
        for (int i=2 ; i<=number ;i++) {  
            if(number%i==0) {  
                c++;  
            }  
        }  
        if(c==1) {  
            System.out.println(number + " : is a prime number");  
        }  
        else {  
            System.out.println(number + " : is not a prime  
number");  
        }  
    }  
}
```

OUTPUT :

Enter your number

11
11 : is a prime number

➤ FIBONACCI SERIES:

```
public class FibonnasiSeries {  
  
    public static void main(String[] args) {  
  
        Scanner input = new Scanner(System.in);  
  
        System.out.println("Enter your number ");  
  
        int number= input.nextInt();  
  
        int a=0;  
        int b=1;  
        int c;  
  
        int i=0;  
  
        System.out.print("FibonnasiSeries is as :");  
  
        while (i<=number) {  
            c=a+b;  
  
            a=b;  
            b=c;  
            i++;  
  
            System.out.print(" " +c);  
        }  
    }  
}
```

OUTPUT :

Enter your number

12

FibonnasiSeries is as : 1 2 3 5 8 13 21 34 55 89 144 233 377

➤ **ARMSTRONG NUMBER:**

```
public class AromstrongNumber {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.println("Enter your number ");  
  
        int number= input.nextInt();  
        int rev =0 ;  
        int rem ;  
        int arms = number ;  
  
        while (number > 0) {  
            rem = number % 10 ;  
  
            rev = rev + rem *rem * rem * rem;  
            number = number / 10 ;  
        }  
        System.out.println(rev);  
  
        if (rev ==arms) {  
            System.out.println(rev + " is a armstrong number");  
        }  
        else {  
            System.out.println(rev+ " is not a armstrong number");  
        }  
    }  
}
```

OUTPUT:

Enter your number

1634

1634

1634 is a armstrong number

➤ SWAP NUMBERS :

```
public class SwapNumber {  
  
    public static void main(String[] args) {  
  
        Scanner input = new Scanner(System.in);  
  
        System.out.println("Enter your number ");  
  
        int number1= input.nextInt();  
  
        int number2= input.nextInt();  
  
  
        int c;  
  
        c=number1;  
  
        number1=number2;  
  
        number2=c;  
  
        System.out.println("The value of number1 is:"+ number1);  
  
        System.out.println("The value of number2 is "+number2);  
  
    }  
}
```

OUTPUT :

Enter your number

10

12

The value of number1 is:12

The value of number2 is 10

➤ Creation of 2 D array by using Scanner class :

```
public class CREATE_2D_ARRAY {  
  
    public static void main(String[] args) {  
  
        Scanner sc=new Scanner(System.in);  
  
        System.out.println("Enter Row Size :");  
  
        int rowsize=sc.nextInt();  
        System.out.println("Enter Column Size :");  
        int colsize=sc.nextInt();  
        int a[][]=new int[rowsize][colsize];  
  
        //for taking values from user  
        System.out.println("Enter the elements :");  
  
        for(int i=0;i<rowsize;i++) {  
            for(int j=0;j<colsize;j++) {  
                a[i][j]=sc.nextInt();  
            }  
  
            //for printing values  
            for(int i=0;i<rowsize;i++) {  
                for(int j=0;j<colsize;j++) {  
                    System.out.println("Value at a["+i+""]["+j+"]th :  
"+a[i][j]);  
                }  
            }  
        }  
    }  
}
```

OUTPUT :

Enter Row Size :

3

Enter Column Size :

2

Enter the elements :

6

5

4

3

2

1

Value at a[0][0]th : 6

Value at a[0][1]th : 5

Value at a[1][0]th : 4

Value at a[1][1]th : 3

Value at a[2][0]th : 2

Value at a[2][1]th : 1

➤ Prime number between range :

```
public class PrimeNumberBetweenRange {  
    public static void main(String[] args) {  
        Scanner jk = new Scanner (System.in);  
        System.out.println("Enter your number?");  
        int number = jk.nextInt();  
        System.out.print("Prime numbers are :");  
        int c=0;  
        for (int i=jk.nextInt(); i<=number ;i++) {  
            for (int j=2 ; j<=i ; j++) {  
                if(i%j==0) {  
                    c++;  
                }  
            }  
            if (c==1) {  
                System.out.print(" "+ i);  
            }  
            else {  
                c=0;  
            }  
        }  
    }  
}
```

OUTPUT:

Enter your number?

100

Prime numbers are :

50

53 59 61 67 71 73 79 83 89 97

➤ Natural Numbers:

```
import java.util.Scanner;

public class NaturalNumbers {

    public static void main(String[] args) {

        Scanner jk = new Scanner (System.in);

        System.out.println("Enter your number?");

        int number = jk.nextInt();

        System.out.print("Natural numbers are : ");

        for (int b=1; b<=number; b++) {

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

        }

    }

}
```

OUTPUT :

Enter your number?

10

Natural numbers are : 1 2 3 4 5 6 7 8 9 10

➤ LEAP YEAR:

```
public class Leap_Year {  
  
    public static void main(String[] args) {  
        // find leap year  
  
        /*condition=  
            1)century proper  divide by 400  OR  
            2) proper divide by 4  
            3) but not divisible by 100*/  
  
        Scanner P=new Scanner(System.in);  
  
        System.out.print("Enter Year=");  
  
        int year= P.nextInt();  
        //int year =2001;  
        if(year% 4==0)  
        {  
            if(year%100==0)  
            {  
                if(year%400==0)  
                {  
                    System.out.println(" leap year");  
                }  
                else {  
                    System.out.println(" not leap year");  
                }  
            }  
            else {  
                System.out.println(" leap year");  
            }  
        }  
        else {  
            System.out.println(" not leap year");  
        }  
    }  
}
```

OUTPUT :

```
Enter Year=  
2021  
not leap year
```


PROGRAMS ON COLLECTIONS:

- Write a program to traverse (or iterate) ArrayList:

```
public class ITERATE_ARRAYLIST {  
  
    public static void main(String args[]) {  
  
        // initialize ArrayList  
        ArrayList<Integer> al = new ArrayList<Integer>();  
  
        // add elements to ArrayList object  
        al.add(3);  
        al.add(17);  
        al.add(6);  
        al.add(9);  
        al.add(7);  
  
        System.out.println("Using Advanced For Loop");  
  
        // printing ArrayList  
        for (Integer num : al) {  
            System.out.println(num);  
        }  
    }  
}
```

OUTPUT:

```
Using Advanced For Loop  
3  
17  
6  
9  
7
```

➤ Write a program to convert List to Array:

```
public class ARRAYLIST_TO_ARRAY_CONVERSION {  
  
    public static void main(String args[]) {  
  
        // Creating and initializing ArrayList  
        ArrayList<String> fruits = new ArrayList<>();  
  
        fruits.add("Apple");  
        fruits.add("Banana");  
        fruits.add("Mango");  
        fruits.add("Pear");  
  
        // ArrayList to String array conversion  
  
        String str [] = new String [fruits.size()];  
  
        for (int i=0 ; i<fruits.size(); i++) {  
  
            str[i]=fruits.get(i);  
  
            System.out.println(str[i]);  
        }  
  
        System.out.println("*****_by using advanced for  
loop_*****");  
        //print element by using advance for loop:  
  
        for ( String ss : str) {  
  
            System.out.println(ss);  
        }  
    }  
}
```

OUTPUT:

```
Apple  
Banana  
Mango  
Pear  
*****_by using advanced for loop_*****  
Apple  
Banana  
Mango  
Pear
```

➤ Write a program to traverse (or iterate) HashSet:

```
import java.util.HashSet;
import java.util.Iterator;

public class ITERATE_HASHSET {

    public static void main(String args[]) {

        // Declaring a HashSet
        HashSet<String> hashset = new HashSet<String>();

        // Add elements to HashSet
        hashset.add("Pear");
        hashset.add("Apple");
        hashset.add("Orange");
        hashset.add("Papaya");
        hashset.add("Banana");

        // Get iterator
        Iterator<String> it = hashset.iterator();

        // Show HashSet elements
        System.out.println("HashSet contains: ");

        while(it.hasNext()) {
            System.out.println(it.next());
        }
    }
}
```

OUTPUT:

```
HashSet contains:
Apple
Pear
Papaya
Orange
Banana
```

- Write a program to convert Array to List:
 - Method 1 : Using Arrays.asList() method

```
import java.util.ArrayList;
import java.util.Arrays;

public class ARRAY_TO_ARRAYLIST_CONVERSION {

    public static void main(String args[]) {

        // Declaring and initializing Array
        String[] cities={"Boston", "Dallas", "New York", "Chicago"};

        //Converting Array to ArrayList using Arrays.asList()
        ArrayList<String> list= new
ArrayList<>(Arrays.asList(cities));

        // Add more elements to the converted list
        list.add("San Francisco");
        list.add("San jose");

        // Print arraylist elements using for-each loop
        for(String s : list) {
            System.out.println(s);
        }
    }
}
```

OUTPUT:

```
Boston
Dallas
New York
Chicago
San Francisco
San jose
```

▪ **Method 2 : Using Collections.addAll() method:**

```
public class ARRAY_TO_ARRAYLIST_CONVERSION2 {  
  
    public static void main(String args[]) {  
        // Creating and initializing Array  
        String[] strArray = {"AAA", "BBB", "CCC", "DDD"};  
  
        // Declaring ArrayList  
        ArrayList<String> al = new ArrayList<>();  
        //Converting Array to ArrayList using addAll() method  
        Collections.addAll(al, strArray);  
  
        // Add more elements to the converted list  
        al.add("YYY");  
        al.add("ZZZ");  
  
        // Displaying arraylist elements using for-each loop  
        for(String s : al) {  
            System.out.println(s);  
        }  
    }  
}
```

OUTPUT:

```
AAA  
BBB  
CCC  
DDD  
YYY  
ZZZ
```

▪ **Method 3 : Using add() method:**

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;

public class ARRAY_TO_ARRAYLIST_CONVERSION3 {

    public static void main(String args[]) {

        // Declaring and instantiating ArrayList in one step
        ArrayList<String> al = new ArrayList();
        // Given initialized array
        String[] strArray = {"Cocacola", "Pepsi", "Fanta", "Dr Pepper"};
        //Converting Array to ArrayList manually
        for (int i=0; i < strArray.length ; i++) {
            // Adding every element of array to the ArrayList
            al.add(strArray[i]);
        }
        // Showing arraylist elements using for-each loop
        for(String str1 : al) {
            System.out.println(str1);
        }
    }
}
```

OUTPUT:

```
Cocacola
Pepsi
Fanta
Dr Pepper
```

➤ Write a program to iterate the HashMap:

```
import java.util.HashMap;

public class ITERATE_HASHMAP {

    public static void main(String args[]) {

        // Creating a HashMap of String keys and String values
        HashMap<String, String> hashmap = new HashMap<String,
String>();

        hashmap.put("Key1", "Value1");
        hashmap.put("Key2", "Value2");

        System.out.println("Iterating or looping map using foreach
loop");

        // Iterating or looping using keySet() method
        for (String key : hashmap.keySet()) {
            System.out.println("key: " + key + " value: " +
hashmap.get(key));
        }
    }
}
```

OUTPUT:

```
Iterating or looping map using foreach loop
key: Key2 value: Value2
key: Key1 value: Value1
```

Write a program to sort ArrayList using Comparable and Comparator?

```
import java.util.TreeSet;

public class TreeSetClass {
    public static void main(String[] args) {
        TreeSet<String> jk = new TreeSet<String>(new
ComparatorDiscussion());
        jk.add("Riddhi");
        jk.add("Siddhi");
        jk.add("Vedant");
        jk.add("Badri");
        jk.add("Digu");
        System.out.println(jk);
    }
}

import java.util.Comparator;

public class ComparatorDiscussion implements Comparator<String> {
    @Override
    public int compare(String o1, String o2) {
        return o1.compareTo(o2);
    }
}
```

OUTPUT :

[Badri, Digu, Riddhi, Siddhi, Vedant]

➤ Write a program to sort ArrayList in descending order:

```
public class ARRAYLIST_DECENDING_ORDER {  
  
    public static void main(String args[]) {  
  
        ArrayList<String> arrList = new ArrayList();  
        arrList.add("Apple");  
        arrList.add("Banana");  
        arrList.add("Pear");  
        arrList.add("Mango");  
  
        /*Unsorted List: ArrayList content before sorting*/  
        System.out.println("ArrayList Before Sorting:");  
  
        for(String s: arrList){  
            System.out.println(s);  
        }  
  
        /* Sorting in decreasing (descending) order*/  
        Collections.sort(arrList, Collections.reverseOrder());  
  
        /* Sorted List in reverse order*/  
        System.out.println("ArrayList in descending order:");  
  
        for(String str: arrList){  
            System.out.println(str);  
        }  
    }  
}
```

OUTPUT:

```
Apple  
Banana  
Pear  
Mango  
ArrayList in descending order:  
Pear  
Mango  
Banana  
Apple
```

➤ **Write a program to convert LinkedList to ArrayList:**

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

public class LINKED_LIST_TO_ARRAYLIST_CONVERSION {

    public static void main(String args[]) {

        // Creating LinkedList Object
        LinkedList<String> linkedlist = new LinkedList<String>();
        linkedlist.add("Mango");
        linkedlist.add("Banana");
        linkedlist.add("Pear");
        linkedlist.add("Apple");
        linkedlist.add("Orange");

        // Converting LinkedList to ArrayList
        List<String> list = new ArrayList(linkedlist);

        for (String s : list) {
            System.out.println(s);
        }
    }
}
```

OUTPUT :

```
Mango
Banana
Pear
Apple
Orange
```

➤ **Write a program to convert HashSet to Array:**

```
import java.util.HashSet;

public class HASHSET_TO_ARRAY_CONVERSION {

    public static void main(String args[]) {

        // Create a HashSet object
        HashSet<String> hashset = new HashSet<String>();

        // Adding elements to HashSet object
        hashset.add("Doctor");
        hashset.add("Engineer");
        hashset.add("Lawyer");
        hashset.add("Police");

        // Printing HashSet elements
        System.out.println("HashSet contains: "+ hashset);

        // Creating an Array of HashSet size
        String[] array = new String[hashset.size()];

        // Converting HashSet to Array using toArray() method
        hashset.toArray(array);

        // Printing Array elements
        System.out.println("Array contains: ");
        for (String str : array) {
            System.out.println(str);
        }
    }
}
```

OUTPUT:

```
HashSet contains: [Engineer, Doctor, Lawyer, Police]
Array contains:
Engineer
Doctor
Lawyer
Police
```

➤ **Write a program to reverse ArrayList in java:**

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

public class REVERSE_ARRAYLIST {

    public static void main(String[] args)
    {
        //Creating an ArrayList object
        ArrayList<String> arrlist = new ArrayList<String>();
        //Adding elements to ArrayList object
        arrlist.add("Apple");
        arrlist.add("Amazon");
        arrlist.add("Facebook");
        arrlist.add("Google");
        arrlist.add("IBM");
        arrlist.add("Tesla");

        //Displaying ArrayList Before Reverse
        System.out.println("Before Reverse ArrayList:");
        System.out.println(arrlist);

        /*Reversing the list using
        Collections.reverse() method*/
        Collections.reverse(arrlist);

        //Displaying list after reverse
        System.out.println("After Reverse ArrayList:");
        System.out.println(arrlist);
    }
}
```

OUTPUT :

```
Before Reverse ArrayList:
[Apple, Amazon, Facebook, Google, IBM, Tesla]
After Reverse ArrayList:
[Tesla, IBM, Google, Facebook, Amazon, Apple]
```

```
public class ITERATE_TREEMAP {
```

- Write a program to iterate TreeMap in java:



```
public static void main(String args[]) {  
  
    // Declaring a TreeMap of String keys and String values  
    TreeMap<String, String> treemap = new TreeMap<String, String>();  
  
    // Add Key-Value pairs to TreeMap  
    treemap.put("Key1", "Pear");  
    treemap.put("Key2", "Apple");  
    treemap.put("Key3", "Orange");  
    treemap.put("Key4", "Papaya");  
    treemap.put("Key5", "Banana");  
  
    // Get Set of entries  
    Set set = treemap.entrySet();  
  
    // Get iterator  
    Iterator it = set.iterator();  
  
    // Show TreeMap elements  
    System.out.println("TreeMap contains: ");  
  
    while(it.hasNext()) {  
        Map.Entry pair = (Entry) it.next();  
        System.out.print("Key is: "+pair.getKey() + " and ");  
        System.out.println("Value is: "+pair.getValue());  
    }  
}
```

OUTPUT:

```
TreeMap contains:  
Key is: Key1 and Value is: Pear  
Key is: Key2 and Value is: Apple  
Key is: Key3 and Value is: Orange  
Key is: Key4 and Value is: Papaya  
Key is: Key5 and Value is: Banana
```