

Sum of N numbers upto n:-

```
Public class sumofNaturalnumbers{  
    Public static void main(String[] args)  
    {  
        int n=10;  
        int sum=(n*(n+1))/2;  
        System.out.println("sum of natural  
        numbers up to '+n+' is: "+sum);  
    }  
}
```

3

The given number is prime number or not:-

```
Public class main{  
    Public static void main(String[]  
        args){  
        int number = 29;  
        boolean isprime = true;  
        if (number <= 1) {
```

```
is prime = false;  
} else {
```

```
for (int i=2; i <= Math.sqrt(number);  
i++) {
```

```
if (number % i == 0) {
```

```
is prime = false;
```

```
break;
```

```
}  
}  
}
```

```
if (isprime)
```

```
system.out.println(number + "is a prime  
number.");
```

```
else
```

```
system.out.println(number + "is not a  
prime number.");
```

```
}
```

```
}
```

3) Factorial of n number:-

```

Public class factorial calculator {
    public static void main (String[] args) {
        int number = 5;
        long factorial = 1;
        for (int i = 1; i <= number; ++i) {
            factorial *= i;
        }
    }
}

```

5 3

```

System.out.println("Factorial of " + num + " = "
                    + factorial);
}

```

3

7) Reverse a number :-

```

Public class ReverseNumber {
    public static void main (String[] args) {
        int number = 12345;
        int reversedNumber = 0;
        while (number != 0) {
            int digit = number % 10;
            reversedNumber = reversedNumber * 10 + digit;
            number /= 10;
        }
    }
}

```



}

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

}

}

5) Armstrong number :-

```
Public class armstrongNumber {
```

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

```
int number = 153;
```

```
int originalnumber, remainder, result = 0;
```

```
original number = number;
```

```
while (originalnumber number != 0) {
```

```
remainder = originalnumber % 10;
```

```
result += Math.Pow(remainder, 3);
```

```
originalnumber /= 10;
```

```
}
```

```
if (result == number)
```

```
System.out.println(number + "is an armstrong  
number.");
```

```
else
```

Happy Number :-

```
Public class HappyNumber {
```

```
    Public boolean is happy (int n) {
```

```
        Set < Integer > seen = new HashSet<>();
```

```
        while (n != 1 && ! seen.contains(n)) {
```

```
            seen.add(n);
```

```
            n = getNext(n);
```

```
        }
```

```
        return n == 1;
```

```
    }
```

no need to use HashSet

```
    Private int getNext (int n) {
```

```
        int totalSum = 0;
```

```
        while (n > 0) {
```

```
            int digit = n % 10;
```

```
            n = n / 10;
```

```
            totalSum += digit * digit;
```

```
        }
```

```
        return totalSum;
```

```
    }
```

7) Palindrome number :-

```
Public class Pallindromenumber {
```

```
    Public static boolean is palindrome  
        (int number)
```

```
    int reversednumber = 0;
```

```
    int originalnumber = number;
```

```
    while (number != 0) {  
        digit int digit = number % 10;  
        reversednumber = reversednumber * 10 + digit;  
        number /= 10;
```

```
    }
```

```
    return originalnumber == reversednumber;
```

```
}
```

Print the val

8) Sum of digits :-

```
Public class sum of Digits {
```

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

```
        int number = 12345;
```

```
        int sum = 0;
```

```
        while (number != 0) {
```

```
            sum += number / 10;
```

```
            number /= 10;
```

```
        }
```



Divisible by 5 and 7 upto n.

```
Public class main{
```

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

```
int n = 100;
```

```
System.out.println("Numbers divisible by 5  
and 7 upto " + n + " : ");
```

```
for (int i = 1; i <= n; i++){
```

```
if (i % 5 == 0 && i % 7 == 0){
```

```
System.out.Print(i + " ");
```

```
}
```

```
}
```

```
}
```

```
}
```

Perfect number upto n :-

```
Public class perfectnumber {
```

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

```
int n = 10000;
```

```
System.out.println("Perfect number up to  
" + n + " : ");
```

```

for (int i=1 ; i<=n ; i++){
    if (isperfect(i)) {
        system.out.println(i);
    }
}
}

```

```

public static boolean isperfect (int num) {

```

```

    int sum=0;

```

```

    for (int i=1 ; i<=num/2 ; i++){

```

```

        if (num % i == 0) {

```

```

            sum += i;

```

```

        }
    }
}

```

11) Fibonacci :-

```

public class fibonacciseries {

```

```

    public static void main (String[] args) {

```

```

        int n=10;

```

```

        int firstterm=0;

```

```

        int secondterm=1;

```



```
System.out.println("Fibonacci series:");
```

```
for (int i=1; i<=n; i++) {
```

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

```
    int nextterm = firstterm + secondterm;
```

```
    firstterm = secondterm;
```

```
    secondterm = nextterm;
```

```
}
```

```
}
```

```
}
```

GCD:-

```
public class GCD {
```

```
    public static int calculateGCD (int num1,
```

```
        int num2) {
```

```
        if (num2 == 0) {
```

```
            return num1;
```

```
        }
```

```
        return calculateGCD(num2, num1 % num2);
```

```
}
```

```
Public Static void main (String[] args.
```

```
int number1 = 48;
```

```
int number2 = 18;
```

```
int gcd = calculateGCD (number1, number2);
```

```
System.out.println("GCD of " + number1 + "  
+ number2 + " is: "
```

```
}
```

```
}
```

13) LCM:-

```
import java.util.Scanner;
```

```
Public class Lcmcalculator {
```

```
Public Static void main (String[] args
```

```
Scanner input = new Scanner (System
```

```
System.out.println("Enter first number");
```

```
int num1 = input.nextInt();
```

```
System.out.println("Enter second number");
```

```
int num2 = input.nextInt();
```

```
int gcd = findGCD(num1, num2);
```

```
int lcm = (num1 * num2) / gcd;
```

```
System.out.println("The lcm of " + num1 +  
    " and " + num2 + " is : " + lcm);
```

```
}
```

```
public static int findGCD(int a, int b){
```

```
    if (b == 0) {
```

```
        return a;
```

```
    }
```

```
    return findGCD(b, a % b);
```

```
}
```

```
}
```

Decimal to Binary conversion:-

```
public class DecimalToBinary {
```

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

```
        int decimal = 10;
```

```
        System.out.println("Decimal: " + decimal);
```

```
        System.out.println("Binary : " + Integer.
```

```
            toBinaryString(decimal));
```

```
    }
```

```
}
```



5) Binary - decimal

```
Public class Binaryto decimal converter {  
    Public static void main (String [] args) {  
        String binarynumber = "101010";  
        int decimalnumber = Integer.parseInt (Binary  
                                                    number);  
        System.out.println ("Binary number: " + binary  
                                                    number);  
        System.out.println ("Decimal number: " +  
                                decimalnumber);  
    }  
}
```

6) ~~de~~ celsius to fahrenheit<sup>h ne</sup>:-

```
Public class celsiustofahrenheit {  
    Public static void main (String [] args) {  
        double celsius = 28.0;  
        double fahrenheit = (celsius * 9/5) + 32;  
    }  
}
```

Fahrenheit to celsius :-

```
Public class FahrenheittoCelsius {
```

```
    Public Static void main (String[] args) {
```

```
        double fahrenheit = 98.6;
```

```
        double celsius = (fahrenheit - 32) * 5/9;
```

```
        System.out.println(fahrenheit + " Fahrenheit  
is equal to " + celsius + " celsius.");
```

```
    }
```

```
}
```

leap year :-

```
import java.util.Scanner;
```

```
Public class leapyear {
```

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

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

```
        System.out.print("enter a year:");
```

```
        int year = input.nextInt();
```

```
        if ((year % 4 == 0 && year % 100 != 0) ||
```

```
            (year % 400 == 0))
```

```

System.out.println(year + " is a leap year.");
} else {
    System.out.println(year + " is not a
    leap year.");
}

```

1a) voting :-

```

Public class voting eligibility {
    Public static void main (String [] args) {
        int age = 18;
        if (age >= 18) {
            System.out.println("you are
            eligible to vote.");
        } else {
            System.out.println("you are not
            eligible to vote yet.");
        }
    }
}

```



sum of square root and cube Root of  
a number.

```
import java . lang . math ;  
public class Rootsumcalculator {  
    public static void main (String[] args) {  
        double number = 64 . 0 ;  
        double SquareRoot = math . sqrt (number) ;  
        double cuberoot = math . cbrt (number)  
        double sum = SquareRoot + cuberoot ;
```

Example  
sample output