**Algorithm**

**Postive Number**

if(number==0)

            System.out.println("Netheir positive nor negative");

        else if(number>0)

            System.out.println("Number is positive");

        else

            System.out.println("Number is negative");

**Odd Even**

        boolean check = (number%2==0) ? true : false;

        if(check)

            System.out.println("No is even");

        else

            System.out.println("No is odd");

**N natural number sum**

 if(n<=0) {System.out.print("No cann't be zero or negative"); return;}

        sum = (n\*(n+1))/2;

        System.out.println("Sum of " + n + " Natural No. is: " + sum);

**N natural number sum in a given range**

if(upperLimit<lowerLimit){

            System.out.println("UpperLimit can't be less than LowerLimit");

        return;}

        int sum= 0;

        for(int i = lowerLimit; i<=upperLimit; i++)

            sum += i;

        System.out.println("Sum of the numbers are: "+ sum);

**Greatest of two numbers**

        if(first>second)

            System.out.println("First Number is greater");

        else if(second>first)

            System.out.println("Second Number is greater");

        else

            System.out.println("Both are equal");

**Greatest of three numbers**

if(first == second || second == third || third == first){

        System.out.print("Please enter three distinct numbers to be compared");

            return;

        }

        if(first > second && second > third)

            System.out.println(first +" is the greatest");

        else if(second> third && third> first)

            System.out.println(second + " is the greatest");

        else

            System.out.println(third + " is the greatest");

**Leap year**

 if((year%4==0 && year%100!=0) || year%400==0)

            System.out.println("Year is leap year");

        else

            System.out.println("Year isn't leap year");

**Prime number**

 for(int i = 2; i<number ; i++){

            if(number%i==0){

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

                return;

            }

        }

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

**Sum of a digit of a number**

int sum=0;

        while(number>0){

            sum+=number%10;

            number=number/10;

        }

        System.out.println(sum + " is the sum of digits");

**Reverse a number**

int reverse=0;

        int rem=0;

        while(number>0){

            rem=number%10;

            reverse = reverse\*10+rem;

            number=number/10;

        }

        System.out.print(reverse + " is reversed number");

**Check weather a no is palindrome or not**

int temp=number;

        int reverse=0;

        int rem;

        while(number>0){

                rem= number%10;

                reverse=reverse\*10+rem;

                number=number/10;

        }

        if(temp==reverse)

            System.out.println("Number is a pallindrome");

        else

            System.out.println("Number is not a pallindrome");

**Amicable Number(Friendly pairs) 🡪 some of proper divisor of first is equal to second number and vice vers**

int count1=0;

        int count2=0;

        for(int i=1;i<=num1/2;i++)

            if(num1%i==0)

                count1=count1+i;

        for(int j=1;j<=num2/2;j++)

            if(num2%j==0)

                count2=count2+j;

        if(count2==num1&& count1==num2)

            System.out.print("Two numbers are Amicable numbers");

        else

            System.out.print("Two numbers are not Amicable numbers");

**Harshad Number 🡪 A no which is completely divisible by some of its digit(153🡪1+5+3=9 🡪 153%9==0)**

int temp=number;

        int sum=0;

        while(temp>0){

            sum+=temp%10;

            temp=temp/10;

        }

        if(number%sum==0)

            System.out.println("Number is Harshad");

        else

            System.out.println("Not a Harshad Number");

**Abundant number 🡪 whose proper divisor’s sum is greater than number itself (12 🡪 1,2,3,4 ,6=6+4+3+2+1=16>12)**

int sum =0;

        for(int i=1;i<=number/2;i++)

            if(number%i==0)

                sum+=i;

        if(number<sum)

            System.out.println("No is an abundant number");

        else

            System.out.println("No is not an abundant number");

**Automorphic Number 🡪 last digit of square of a number is equal to no itself 🡪 25 =625🡪 last digits 25=original number**

  int temp = num;

    int square = num\*num;

    int flag=0;

    while(num>0){

        if(num%10!=square%10){

            flag=1; break;}

        num=num/10;

        square=square/10;

    }

    if(flag==1)

        System.out.println("Not automorphic");

    else

        System.out.println("Automorphic Number");

**Fibonacci Numbers**

 int currentTerm=1;

        int previousTerm=0;

        int temp=0;

        System.out.print(previousTerm + " " + currentTerm + " ");

        for(int i=2;i<num;i++){

            temp=currentTerm+previousTerm;

            previousTerm=currentTerm;

            currentTerm=temp;

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

**Factorial of a number**

 if(num<0){System.out.println("Number cann't be Negative"); return;}

        long fact=1;

        for(int i=num;i>1;i--){

            fact\*=i;

        }

        System.out.println("The factorial of the no is: " + fact);

**Power of a number + use of format specifiers**

 System.out.print("Enter the base number: ");

        double x = sc.nextDouble();

        System.out.print("Enter the Exponential number: ");

        double y =sc.nextDouble();

        System.out.printf("pow(%.3f, %.3f) is %.3f%n", x, y, Math.pow(x, y));

**Factors of a number :**

 System.out.print("Factors are: ");

        for(int i=1;i<=number;i++){

            if(number%i==0)

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

        }

**Strong Number:- 145 🡪 1!+4!+5!=145 so 145 is a strong number.**

 int temp = number;

        int sum=0;

        int rem=0;

        while(temp>0){

            rem=temp%10;

            int fact=1;

            for(int i=rem;i>0;i--)

                fact\*=i;

            sum+=fact;

            temp=temp/10;

        }

        if(number==sum)

            System.out.print(number + " is a strong number");

        else

            System.out.print(number + " not is a strong number");

**Perfect Number :- sum of proper factor becomes equal to a number 28🡪 factor 1+2+4+7+14=28**

 int sum=1;

        for(int i=2;i<=number/2;i++){

            if(number%i==0)

                sum+=i;

        }

        if(sum==number)

            System.out.print(number + " is a perfect number");

**LCM OF TWO NUMBERS :-**

int firstNumber = sc.nextInt();

        int secondNumber = sc.nextInt();

        int largerNumber = (firstNumber>secondNumber)

                            ? firstNumber : secondNumber;

        int lcm = firstNumber\* secondNumber;

        for(int i = lcm; i >=largerNumber ; i--){

            if(i%firstNumber==0 && i%secondNumber==0)

                lcm=i;

        }

        System.out.println(lcm + " is the lcm of the numbers");

        sc.close(); //Not necessary but good practice! why? -->

**HCF/GCD OF TWO NUMBER :-**

int firstNumber = sc.nextInt();

        int secondNumber = sc.nextInt();

        int smallerNumber = (firstNumber<secondNumber)

                            ? firstNumber : secondNumber;

        int hcf = smallerNumber;

        for(int i = hcf; i >=1 ; i--){

            if(firstNumber%i==0 && secondNumber%i==0){

                hcf=i;

                break;

            }

        }

        System.out.println(hcf + " is the hcf/gcd of the numbers");

**Bcd to decimal :-**

  int num = scanner.nextInt();

        int base=1;

        int rem= 0;

        int decimalValue = 0;

        while(num>0){

            rem = num%10;

            decimalValue += base\*rem;

            base = base\*2;

            num=num/10;

        }

        System.out.println(decimalValue + " is the decimal value of given binary");

       /\*USING THE BUILD IN FUNTION SO AS TO CONVERT BINARY TO DECIMAL \*/

        // String binaryNumber = "1001";

        // System.out.println(Integer.parseInt(binaryNumber, 2));

**Bcd to octal :-**

 int bcd = scanner.nextInt();

        int rem =0;

        int base=1;

        int decimalValue=0;

        int octalValue=0;

        while(bcd>0){

            rem = bcd%10;

            decimalValue += base\*rem;

            base = base\*2;

            bcd=bcd/10;

        }

        rem=0;

        base=1;

        while(decimalValue>0){

            rem = decimalValue%8;

            octalValue += rem\*base;

            decimalValue = decimalValue/8;

            base = base\*10;

        }

        System.out.println(octalValue + " is the octal value of given bcd");

**Decimal to BCD :-**

 int decimal = scanner.nextInt();

        int base=1;

        int rem=0;

        int binaryValue=0;

        while(decimal>0){

            rem = decimal%2;

            binaryValue += base\*rem;

            decimal = decimal/2;

            base = base\*10;

        }

        System.out.println(binaryValue + " is the binary value of given decimal value");

**Decimal to octal :-**

      int decimal = scanner.nextInt();

        int rem = 0;

        int base = 1;

        int octalValue = 0;

        while(decimal>0){

            rem= decimal%8;

            octalValue += base\*rem;

            decimal=decimal/8;

            base = base\*10;

        }

        System.out.println(octalValue + " is the octal equivalent for decimal number");