1. Write a Java Program to find GCD of two given numbers.

import **java.util.Scanner**;

**public** **class** Ass1\_1 {

**static** **int** gcd(**int** n1,**int** n2)

    {

        if(n2==0)

            return n1;

        return gcd(n2,n1%n2);

    }

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

**Scanner** sc = new Scanner(System.in);

*//int n1 = sc.nextInt();*

*//int n2 = sc.nextInt();*

        System.out.println("GCD : "+gcd(4,6));

    }

}

1. Write a java program to LCM of TWO given number.

import java**.**util**.**Scanner;

**public** **class** Ass1\_2 {

**static** **int** gcd(**int** n1,**int** n2)

    {

        if(n2==0)

            return n1;

        return gcd(n2,n1%n2);

    }

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

        Scanner sc = new Scanner(System.in);

**int** n1 = sc.nextInt();

**int** n2 = sc.nextInt();

**int** gcd=gcd(n1,n2);

**int** lcm=(n1\*n2)/gcd;

        System.out.println(lcm);

*/\**

*\**

*\* 4=2\*2*

*\* 6=2\*3*

*\* lcm=2\*2\*3*

*\**

*\*/*

    }

}

1. Write a Java Program to print all the Prime Factorsof the Given Number.

**public** **class** Ass1\_3 {

**static** **void** prime(**int** n, **int** i)

    {

        if(n==1)

            return ;

        if(n%i==0)

        {

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

            n/=i;

        }

        else{

            i++;

        }

        prime(n, i);

    }

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

        prime(6,2);

    }

}

1. Check whether the Given Numberis a Palindrome or NOT.

**public** **class** Ass1\_4 {

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

**int** number=123;

**boolean** b = isPalindrome(number);

        if(b)

        {

            System.out.println("It is palindrome.......");

        }

        else

        {

            System.out.println("It is not palindrome.......");

        }

    }

**private** **static** **boolean** isPalindrome(**int** num) {

**int** rev=reverseNumber(num);

        return num==rev;

    }

**private** **static** **int** reverseNumber(**int** num)

    {

        return reverseNumber(num,0);

    }

**private** **static** **int** reverseNumber(**int** num, **int** rev) {

       if(num==0)

            return rev;

        return reverseNumber(num/10, rev\*10+num%10);

    }

}

1. Write a Java Program to check whether the Given Number is Prime Number or NOT.

import java**.**util**.**Scanner;

**public** **class** Ass1\_8 {

**public** **static** **void** main(String[] args)

    {

        Scanner sc = new Scanner(System.in);

**int** number = sc.nextInt();

        if (isPrime(number, number / 2)) {

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

        } else {

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

        }

    }

**public** **static** **boolean** isPrime(**int** number, **int** divisor) {

        if (number <= 1) {

            return false;

        }

        if (divisor == 1) {

            return true;

        }

        if (number % divisor == 0) {

            return false;

        }

        return isPrime(number, divisor - 1);

    }

}

1. Write a Java Program to check whether the given number is Armstrong Numberor NOT.

**public** **class** Ass1\_6 {

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

**int** num = 123;

        if(isArmstrong(num))

        {

            System.out.println("It is Armstrong ");

        }

        else

        {

            System.out.println("It is not Armstrong ");

        }

    }

**private** **static** **boolean** isArmstrong (**int** num) {

**int** rev=reverseNumber(num);

        return num==rev;

    }

**private** **static** **int** reverseNumber(**int** num)

    {

        return reverseNumber(num,0);

    }

**private** **static** **int** reverseNumber(**int** num, **int** rev) {

       if(num==0)

            return rev;

        return reverseNumber(num/10, rev+(**int**)Math.pow((num%10), 3));

    }

}

1. Write a Java Program to check whether the given number is Perfect Numberor NOT.

**public** **class** Ass1\_7 {

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

**int** num = 6;

        if(isPerfect(num))

        {

            System.out.println("It is perfect number ");

        }

        else

        {

            System.out.println("It is not perfect number ");

        }

    }

**public** **static** **boolean** isPerfect(**int** number) {

        if (number <= 1)

            return false;

        return number == sumOfFactors(number, 1);

    }

**public** **static** **int** sumOfFactors(**int** number, **int** d) {

        if (d > number / 2)

            return 0;

        if (number % d == 0)

            return d + sumOfFactors(number, d + 1);

        else

            return sumOfFactors(number, d + 1);

    }

}

1. Write a Java Program to check whether the given numbers are Amicable Numbers or NOT.

**public** **class** Ass1\_9 {

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

**int** num1 = 220;

**int** num2 = 284;

        if(isAmicableNumber(num1,num2))

        {

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

        }

        else

        {

            System.out.println("Not Amicable Number");

        }

    }

**private** **static** **boolean** isAmicableNumber(**int** num1, **int** num2) {

**int** sum1 = itsDivisorSum(num1,num1-1);

**int** sum2 = itsDivisorSum(num2,num1-1);

        return sum1 == num2 && sum2 == num1 && num1 != num2;

    }

**private** **static** **int** itsDivisorSum(**int** num1, **int** divisor) {

        if( divisor == 0 )

            return 0;

        if( num1 % divisor == 0 )

            return divisor + itsDivisorSum(num1, divisor-1);

        return itsDivisorSum(num1, divisor-1);

    }

}