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

**import java.util.\*;**

**public class GCD {**

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

**if (b == 0) {**

**return a;**

**} else {**

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

**}**

**}**

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

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

**System.out.println("enter first number :");**

**int num1 =sc.nextInt();**

**System.out.println("enter second number :");**

**int num2 = sc.nextInt();**

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

**System.out.println("GCD is : " + gcd);**

**}**

**}**

**=====================================================================================**

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

**import java.util.\*;**

**public class LCMCalculator {**

**public static int calculateGCD(int a, int b) {**

**if (b == 0) {**

**return a;**

**} else {**

**return calculateGCD(b, a % b);**

**}**

**}**

**private static int calculateLCM(int a, int b) {**

**return (a \* b) / calculateGCD(a, b);**

**}**

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

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

**System.out.println("enter first number :");**

**int num1 =sc.nextInt();**

**System.out.println("enter second number :");**

**int num2 = sc.nextInt();**

**int lcm = calculateLCM(num1, num2);**

**System.out.println("LCM is: " + lcm);**

**}**

**}**

=========================================================================================

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

**import java.util.Scanner;**

**public class PrimeFactor{**

**static boolean isPrime(int n, int i) {**

**if (n <= 2)**

**return (n == 2) ? true : false;**

**if (n % i == 0)**

**return false;**

**if (i \* i > n)**

**return true;**

**return isPrime(n, i + 1);**

**}**

**// Function to print prime factors recursively**

**static void primeFactors(int n, int i) {**

**if (n <= 1)**

**return;**

**if (n % i == 0 && isPrime(i, 2)) {**

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

**primeFactors(n / i, i);**

**} else {**

**primeFactors(n, i + 1);**

**}**

**}**

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

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

**System.out.print("Enter a number: ");**

**int num = scanner.nextInt();**

**scanner.close();**

**System.out.print("Prime factors of " + num + " are: ");**

**primeFactors(num, 2);**

**}**

**}**

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

**public class PalindromeCheck {**

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

**int number = 12321; // Example number, change this to test different numbers**

**if (isPalindrome(number))**

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

**else**

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

**}**

**// Function to check if a number is palindrome recursively**

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

**// Call recursive helper function with number, original number, and 0 as initial index**

**return isPalindromeHelper(number, number, 0);**

**}**

**// Recursive helper function to check if a number is palindrome**

**private static boolean isPalindromeHelper(int original, int reversed, int index) {**

**// Base case: If index reaches the middle of the number, return true**

**if (index == original / 10)**

**return original == reversed;**

**// Extract the digit at the current index**

**int digit = reversed % 10;**

**// Shift the reversed number to the right**

**reversed /= 10;**

**// Recursive call: Check next digit and increment index**

**return isPalindromeHelper(original, reversed, index + 1) && (digit == original % 10);**

**}**

**}**

**==============================================================================**

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

**public class PrimeChecker {**

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

**int number = 19; // Example number to check**

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

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

**} else {**

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

**}**

**}**

**// Function to check prime recursively**

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

**if (i == 1) {**

**return true;**

**} else {**

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

**return false;**

**} else {**

**return isPrime(number, i - 1);**

**}**

**}**

**}**

**}**

**==============================================================================**

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

**import java.util.Scanner;**

**public class ArmstrongNumber {**

**// Function to calculate the power of a number**

**static int power(int num, int n) {**

**if (n == 0)**

**return 1;**

**return num \* power(num, n - 1);**

**}**

**// Function to count the number of digits in a number**

**static int countDigits(int num) {**

**if (num == 0)**

**return 0;**

**return 1 + countDigits(num / 10);**

**}**

**// Function to check whether the number is Armstrong or not**

**static boolean isArmstrong(int num, int n) {**

**int digits = countDigits(num);**

**int sum = 0;**

**int temp = num;**

**while (temp > 0) {**

**int digit = temp % 10;**

**sum += power(digit, digits);**

**temp /= 10;**

**}**

**return num == sum;**

**}**

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

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

**System.out.print("Enter a number: ");**

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

**scanner.close();**

**if (isArmstrong(number, countDigits(number)))**

**System.out.println(number + " is an Armstrong number.");**

**else**

**System.out.println(number + " is not an Armstrong number.");**

**}**

**}**

**===================================================================================**

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

**public class PerfectNumber {**

**// Function to check if a number is perfect or not**

**static boolean isPerfect(int num, int divisor, int sum) {**

**if (divisor == num) {**

**// If the sum of divisors is equal to the number itself, it's a perfect number**

**return sum == num;**

**} else {**

**// If the current divisor divides the number, add it to the sum**

**if (num % divisor == 0) {**

**sum += divisor;**

**}**

**// Recursively check for next divisor**

**return isPerfect(num, divisor + 1, sum);**

**}**

**}**

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

**int number = 28; // Example number to check**

**if (isPerfect(number, 1, 0)) {**

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

**} else {**

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

**}**

**}**

**}**

**=====================================================================================**

1. **Write a Java Program to check whether the given numbers are Amicable Numbersor NOT.**

**import java.util.Scanner;**

**public class ANumbers {**

**// Function to calculate the sum of proper divisors of a number**

**public static int sumOfDivisors(int num, int divisor) {**

**if (divisor == 1) {**

**return 1;**

**}**

**if (num % divisor == 0) {**

**return divisor + sumOfDivisors(num, divisor - 1);**

**}**

**return sumOfDivisors(num, divisor - 1);**

**}**

**// Function to check if two numbers are amicable**

**public static boolean areAmicable(int num1, int num2) {**

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

**int sum2 = sumOfDivisors(num2, num2 - 1);**

**return (sum1 == num2 && sum2 == num1);**

**}**

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

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

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

**int num1 = scanner.nextInt();**

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

**int num2 = scanner.nextInt();**

**if (areAmicable(num1, num2)) {**

**System.out.println(num1 + " and " + num2 + " are amicable numbers.");**

**} else {**

**System.out.println(num1 + " and " + num2 + " are not amicable numbers.");**

**}**

**scanner.close();**

**}**

**}**

**=====================================================================================**

1. **Write a Java Program to check whether the given number is Ramanujam's Numberor NOT.**

**public class RNumber {**

**// Function to check if a number is a Ramanujan number**

**public static boolean isRamanujan(int num, int a, int b, int c, int d) {**

**// Base case: If num is 0, return true**

**if (num == 0)**

**return true;**

**// If all possible combinations are checked**

**if (a > num || b > num || c > num || d > num)**

**return false;**

**// Recursively check for different combinations**

**if (a \* a \* a + b \* b \* b == num && c \* c \* c + d \* d \* d == num)**

**return true;**

**else if (a \* a \* a + b \* b \* b > num)**

**return isRamanujan(num, a, b - 1, c, d);**

**else if (c \* c \* c + d \* d \* d > num)**

**return isRamanujan(num, a, b, c, d - 1);**

**else**

**return isRamanujan(num, a + 1, b, c, d);**

**}**

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

**int number = 1729; // Change this number to test different numbers**

**if (isRamanujan(number, 1, (int) Math.cbrt(number), 1, (int) Math.cbrt(number)))**

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

**else**

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

**}**

**}**

**=====================================================================================**

1. **Write a Java Program check whether the given number is Automorphic Numberor NOT.**

**public class ANumber {**

**// Function to check if a number is automorphic**

**static boolean isAutomorphic(int num, int square) {**

**if (num == 0) {**

**return true;**

**}**

**// Extract last digit of number and last digit of square**

**int digitNum = num % 10;**

**int digitSquare = square % 10;**

**// If last digits don't match, number is not automorphic**

**if (digitNum != digitSquare) {**

**return false;**

**}**

**// Recursively check with next digits**

**return isAutomorphic(num / 10, square / 10);**

**}**

**// Function to compute square of a number**

**static int square(int num) {**

**return num \* num;**

**}**

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

**int number = 76;**

**// Calculate the square of the number**

**int squareNum = square(number);**

**// Check if the number is automorphic**

**if (isAutomorphic(number, squareNum)) {**

**System.out.println(number + " is an automorphic number.");**

**} else {**

**System.out.println(number + " is not an automorphic number.");**

**}**

**}**

**}**