

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

```
1 package com.gcd.in;
2
3 import java.util.Scanner;
4
5 public class Program {
6
7     public static int calcGcd(int num1, int num2) {
8         if (num2 == 0) {
9             return num1;
10        } else {
11            return calcGcd(num2, num1 % num2);
12        }
13    }
14
15    public static void main(String[] args) {
16        Scanner sc = new Scanner(System.in);
17        System.out.println("Enter the first number : ");
18        int num1 = sc.nextInt();
19        System.out.println("Enter the Second number : ");
20        int num2 = sc.nextInt();
21
22        int num3 = calcGcd(num1, num2);
23        sc.close();
24
25        System.out.println("Greatest common divisor of " + num1 + " and " + num2 + " is : " + num3);
26    }
27 }
```

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

```
src > com > LCM > in > J Program.java > {} com.LCM.in
1 package com.LCM.in;
2
3 import java.util.Scanner;
4
5 public class Program {
6
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         System.out.println("Enter the first number : ");
10        int num1 = sc.nextInt();
11
12        System.out.println("Enter the first number : ");
13        int num2 = sc.nextInt();
14
15        int n1 = num1;
16        int m1 = num2;
17        while( m1 != n1 ){
18            if( m1 > n1 )
19                n1 += num1;
20            else
21                m1 += num2;
22        }
23        sc.close();
24        System.out.println( "LCM of "+num1+" and "+num2+" is " + m1 );
25    }
26 }
27
28 }
29
```

3. Write a Java Program to print all the Prime Factors of the Given Number.

```
src > com > primefactor > in > J Program.java > ...
1  package com.primefactor.in;
2
3
4  import java.util.Scanner;
5
6  public class Program {
7
8      public static int primeFactors(int n){
9          int i = 2;
10         while (i <= n) {
11             if (n % i == 0) {
12                 System.out.print(i + " ");
13                 n = n / i;
14             } else {
15                 i++;
16             }
17         }
18         return i;
19     }
20
21
22
23     Run | Debug
24     public static void main(String[] args)
25     {
26         Scanner sc = new Scanner(System.in);
27         System.out.println(x:"Give me a number : ");
28         int n = sc.nextInt();
29         primeFactors(n);
30         sc.close();
31     }
32 }
```

4. Check whether the Given Number is a Palindrome or NOT.

```
src > com > palindrome > in > J Program.java > Program > main(String[])
1  package com.palindrome.in;
2
3  import java.util.Scanner;
4
5
6  public class Program {
7
8      private static boolean isPalindrome(String input) {
9          int left = 0;
10         int right = input.length() - 1;
11
12         while (left < right) {
13             if (input.charAt(left) != input.charAt(right)) {
14                 return false;
15             }
16             left++;
17             right--;
18         }
19
20         return true;
21     }
22
23
24     Run | Debug
25     public static void main(String[] args) {
26         Scanner sc = new Scanner(System.in);
27         System.out.print(s:"Enter a string: ");
28         String input = sc.nextLine();
29         boolean isPalindrome = isPalindrome(input);
30         if (isPalindrome) {
31             System.out.println(x:"The given string is a palindrome.");
32         } else {
33             System.out.println(x:"The given string is not a palindrome.");
34         }
35         sc.close();
36     }
37 }
```

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

```
src > com > primenumber > in > J Program.java > {} com.primenumber.in
1 package com.primenumber.in;
2
3 import java.util.Scanner;
4
5 public class Program {
6
7     static boolean isPrime(int num, int divisor) {
8         if (num <= 1) {
9             return false;
10        }
11        if (divisor > Math.sqrt(num)) {
12            return true;
13        }
14        if (num % divisor == 0) {
15            return false;
16        }
17        return isPrime(num, divisor + 1);
18    }
19
20    Run | Debug
21    public static void main(String[] args) {
22        Scanner sc = new Scanner(System.in);
23        System.out.print(s:"Enter a number: ");
24        int number = sc.nextInt();
25        if (isPrime(number, divisor:2)) {
26            System.out.println(number + " is a prime number");
27        } else {
28            System.out.println(number + " is not a prime number");
29        }
30        sc.close();
31    }
32 }
```

6. Write a Java Program to check whether the given number is Armstrong Number or NOT.

```
src > com > armstrongno > in > J Program.java > {} com.armstrongno.in
1 package com.armstrongno.in;
2
3 import java.util.Scanner;
4 import java.lang.Math;
5
6 public class Program {
7
8     public static boolean isArmstrong(int number) {
9         int sum = 0;
10        int originalNumber = number;
11
12        while (number != 0) {
13            int digit = number % 10;
14            sum += Math.pow(digit, 3);
15            number /= 10;
16        }
17
18        return sum == originalNumber;
19    }
20
21    Run | Debug
22    public static void main(String[] args) {
23        Scanner scanner = new Scanner(System.in);
24
25        System.out.print(s:"Enter a 3-digit number: ");
26        int inputNumber = scanner.nextInt();
27
28        if (inputNumber >= 100 && inputNumber <= 999) {
29            if (isArmstrong(inputNumber)) {
30                System.out.println(inputNumber + " is an Armstrong number.");
31            } else {
32                System.out.println(inputNumber + " is not an Armstrong number.");
33            }
34        } else {
35            System.out.println(x:"Please enter a valid 3-digit number.");
36        }
37        scanner.close();
38    }
39 }
```

7. Write a Java Program to check whether the given number is Perfect Number or NOT.

```
src > com > perfectnumber > in > J Program.java > {} com.perfectnumber.in
1 package com.perfectnumber.in;
2
3 import java.util.Scanner;
4
5 public class Program {
6
7     static int isPerfectNum(int num, int fact, int sum) {
8         if (fact >= num) {
9             return sum;
10        }
11
12        else {
13            if (num % fact == 0) {
14                sum = sum + fact;
15            }
16        }
17        return isPerfectNum(num, fact + 1, sum);
18    }
19
20    Run | Debug
21    public static void main(String args[]) {
22        Scanner sc = new Scanner(System.in);
23        System.out.println(x:"Enter a number : ");
24        int num = sc.nextInt();
25        int fact = 1;
26        int sum = 0;
27        int perfNum = isPerfectNum(num, fact, sum);
28
29        if (num == perfNum)
30            System.out.println(x:"Perfect number");
31        else
32            System.out.println(x:"Not Perfect number");
33
34        sc.close();
35    }
36 }
```

9. Write a Java Program check whether the given number is Automorphic Number or NOT.

```
src > com > automorphic > in > J Program.java > {} com.automorphic.in
1 package com.automorphic.in;
2
3 import java.util.Scanner;
4
5 public class Program {
6
7     private static void Automorphic(int a) {
8
9         if ((a % 10) == (((int) Math.pow(a, 2)) % 10)) {
10             System.out.println(x:" It is Automorphic number");
11         } else {
12             System.out.println(x:" It is not a Automorphic number");
13         }
14     }
15
16    Run | Debug
17    public static void main(String[] args) {
18        Scanner sc = new Scanner(System.in);
19        System.out.println(x:"Enter a number : ");
20        int a = sc.nextInt();
21        Automorphic(a);
22        sc.close();
23    }
24 }
```

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

```
src > com > amicable > no > J Program.java > ...
1  package com.amicable.no;
2
3  import java.util.Scanner;
4
5  public class Program {
6
7      private static boolean isAmicable(int num1, int num2) {
8          int sum1 = sumOfDivisors(num1);
9          int sum2 = sumOfDivisors(num2);
10         if (sum1 == num2 && sum2 == num1 && num1 != num2) {
11             return true;
12         }
13         return false;
14     }
15
16     private static int sumOfDivisors(int num) {
17         int sum = 0;
18         for (int i = 1; i <= num / 2; i++) {
19             if (num % i == 0) {
20                 sum += i;
21             }
22         }
23         return sum;
24     }
25
26
27     Run | Debug
28     public static void main(String[] args) {
29         Scanner sc = new Scanner(System.in);
30         System.out.print(s:"Enter the first number: ");
31         int num1 = sc.nextInt();
32
33         System.out.print(s:"Enter the second number: ");
34         int num2 = sc.nextInt();
35
36         boolean isAmicable = isAmicable(num1, num2);
37         if (isAmicable) {
38             System.out.println(x:"The given numbers are Amicable Numbers.");
39         } else {
40             System.out.println(x:"The given numbers are not Amicable Numbers.");
41         }
42     }
43 }
44
45
```