

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

```
package com.example.main;

import java.util.Scanner;

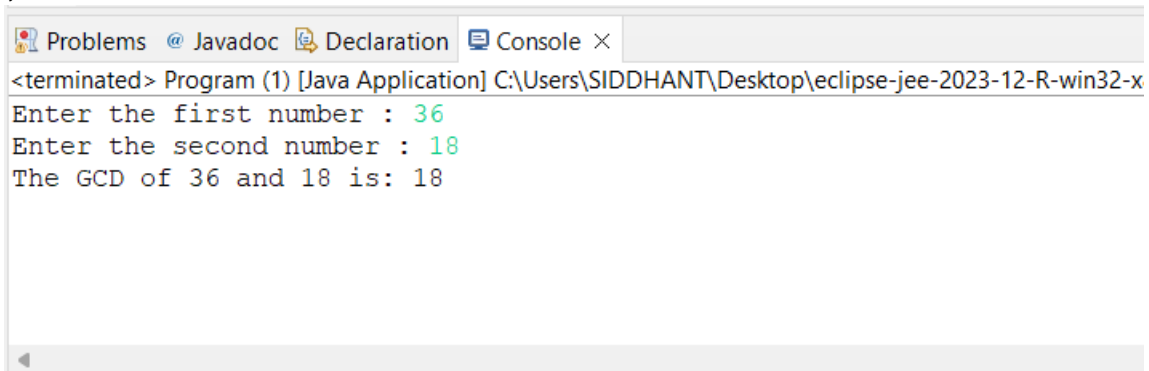
public class Program {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the first number : ");

        int number1 = sc.nextInt();

        System.out.print("Enter the second number : ");
        int number2 = sc.nextInt();

        int gcd = findGCD(number1, number2);
        System.out.println("The GCD of " + number1 + " and " + number2
+ " is: " + gcd);
    }

    public static int findGCD(int a, int b) {
        if (b == 0) {
            return a;
        } else {
            return findGCD(b, a % b);
        }
    }
}
```

A screenshot of the Eclipse IDE's console window. The window has tabs for 'Problems', '@ Javadoc', 'Declaration', and 'Console'. The 'Console' tab is active, showing the output of the Java program. The text in the console is: '<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x', 'Enter the first number : 36', 'Enter the second number : 18', and 'The GCD of 36 and 18 is: 18'. The numbers 36 and 18 are highlighted in green, and the result 18 is also highlighted in green. The console window has a scrollbar at the bottom.

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

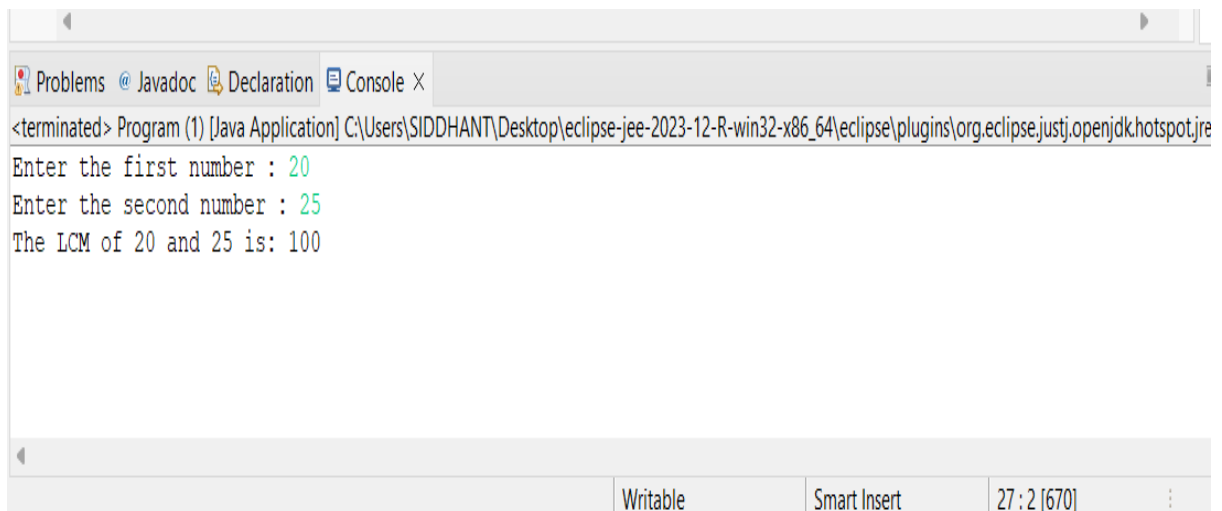
```
package com.example.main;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the first number : ");
        int number1 = sc.nextInt();
        System.out.print("Enter the second number : ");
        int number2 = sc.nextInt();
        int lcm = findLCM(number1, number2);
        System.out.println("The LCM of " + number1 + " and " + number2
+ " is: " + lcm);
    }

    public static int findLCM(int a, int b) {
        return (a * b) / findGCD(a, b);
    }

    public static int findGCD(int a, int b) {
        if (b == 0) {
            return a;
        } else {
            return findGCD(b, a % b);
        }
    }
}
```



The screenshot shows the Eclipse IDE's console window. The title bar indicates the file path: C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre. The console output shows the program's execution: it prompts for the first number (20) and the second number (25), and then displays the result: "The LCM of 20 and 25 is: 100". The status bar at the bottom shows "Writable", "Smart Insert", and the cursor position "27 : 2 [670]".

```
<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre
Enter the first number : 20
Enter the second number : 25
The LCM of 20 and 25 is: 100
```

Writable Smart Insert 27 : 2 [670]

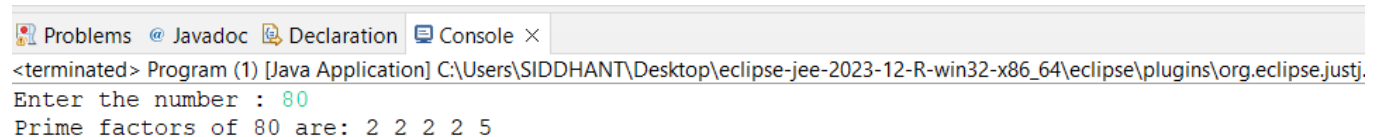
3. Write a Java Program to print all the Prime Factors of the Given Number.

```
package com.example.main;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number = sc.nextInt();
        System.out.print("Prime factors of " + number + " are: ");
        printPrimeFactors(number);
    }

    public static void printPrimeFactors(int n) {
        while (n % 2 == 0) {
            System.out.print(2 + " ");
            n /= 2;
        }
        for (int i = 3; i <= Math.sqrt(n); i += 2) {
            while (n % i == 0) {
                System.out.print(i + " ");
                n /= i;
            }
        }
        if (n > 2) {
            System.out.print(n);
        }
    }
}
```

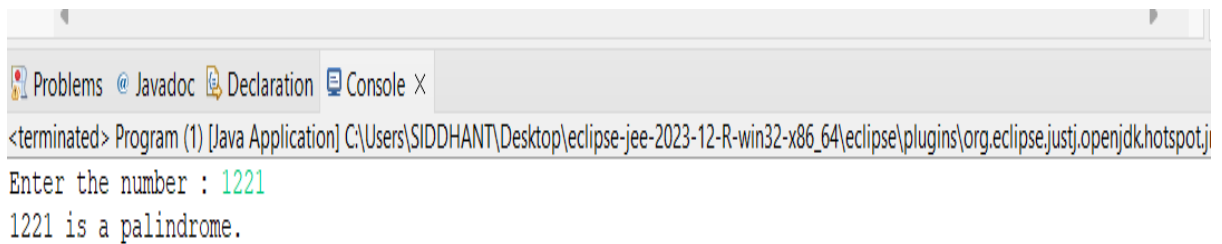


The screenshot shows the Eclipse IDE's console window. The title bar includes tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output shows the program's execution: it prompts 'Enter the number : 80' and then prints 'Prime factors of 80 are: 2 2 2 2 5'. The file path in the title bar is 'C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.'

```
<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.
Enter the number : 80
Prime factors of 80 are: 2 2 2 2 5
```

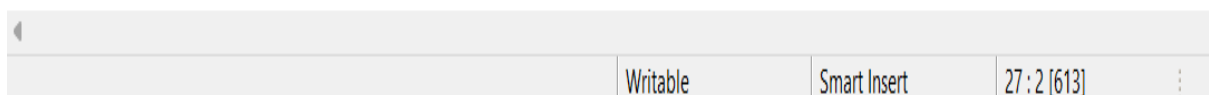
4. Check whether the Given Number is a Palindrome or NOT.

```
public class Program {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter the number : ");  
        int number = sc.nextInt();  
        if (isPalindrome(number)) {  
            System.out.println(number + " is a palindrome.");  
        } else {  
            System.out.println(number + " is not a palindrome.");  
        }  
    }  
  
    public static boolean isPalindrome(int number) {  
        int originalNumber = number;  
        int reverse = 0;  
        while (number != 0) {  
            int lastDigit = number % 10;  
            reverse = reverse * 10 + lastDigit;  
            number /= 10;  
        }  
        return originalNumber == reverse;  
    }  
}
```



The screenshot shows the Eclipse IDE's console window. The title bar includes 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output shows the program has terminated, followed by the prompt 'Enter the number : 1221' and the output '1221 is a palindrome.'.

```
<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.j  
Enter the number : 1221  
1221 is a palindrome.
```



The screenshot shows the Eclipse IDE's status bar at the bottom. It contains four sections: 'Writable', 'Smart Insert', '27 : 2 [613]', and a vertical ellipsis icon.

Writable	Smart Insert	27 : 2 [613]	⋮
----------	--------------	--------------	---

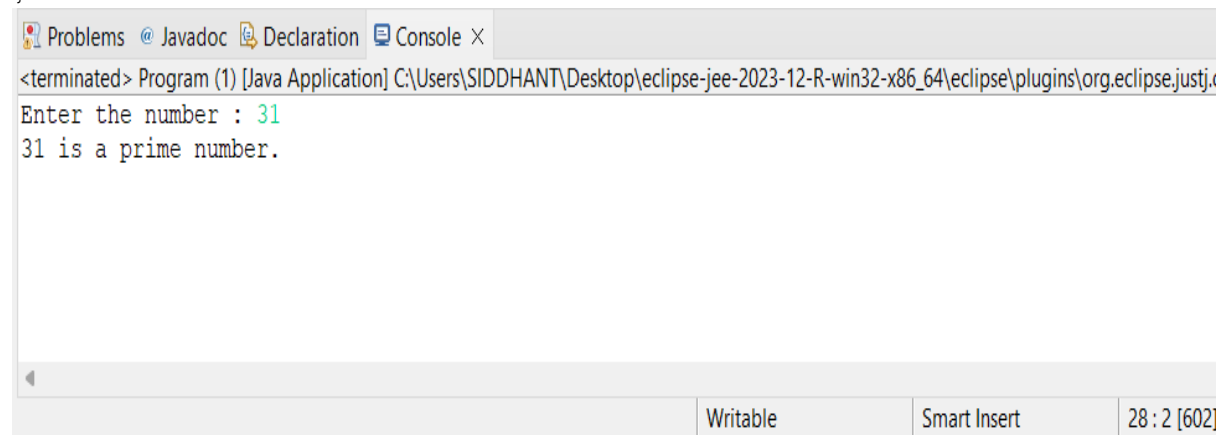
5. Write a Java Program to check whether the Given Number is Prime Number or NOT.

```
package com.example.main;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number = sc.nextInt();
        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
    }

    public static boolean isPrime(int n) {
        if (n < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```



Problems @ Javadoc Declaration Console ×

<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.
Enter the number : 31
31 is a prime number.

Writable Smart Insert 28 : 2 [602]

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

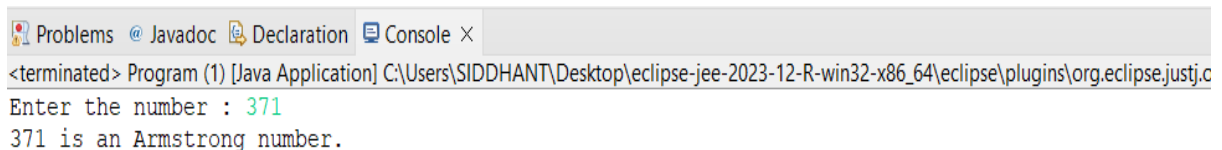
```
package com.example.main;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number = sc.nextInt(); // 153
        if (isArmstrong(number)) {
            System.out.println(number + " is an Armstrong number.");
        } else {
            System.out.println(number + " is not an Armstrong
number.");
        }
    }

    // Method to check if a number is an Armstrong number
    public static boolean isArmstrong(int number) {
        int originalNumber, remainder, result = 0, n = 0;

        originalNumber = number;
        // Finding the number of digits
        while (originalNumber != 0) {
            originalNumber /= 10;
            ++n;
        }
        originalNumber = number;
        // Checking if the number is Armstrong
        while (originalNumber != 0) {
            remainder = originalNumber % 10;
            result += Math.pow(remainder, n);
            originalNumber /= 10;
        }
        if (result == number) {
            return true;
        } else {
            return false;
        }
    }
}
```



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output of the Java program is displayed, showing the prompt 'Enter the number : 371' and the result '371 is an Armstrong number.' The console title bar indicates the program is terminated and running in a Java Application environment.

```
<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.c
Enter the number : 371
371 is an Armstrong number.
```

Writable

Smart Insert

40 : 2 [1015

```
package com.example.main;

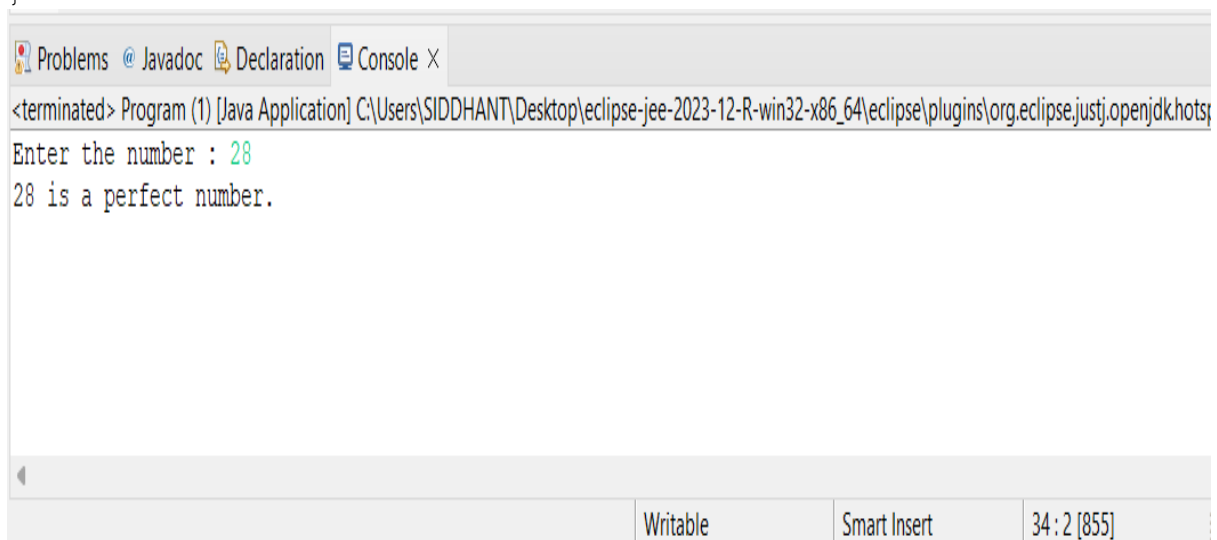
import java.util.Scanner;

public class Program {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number = sc.nextInt(); // 28
        if (isPerfectNumber(number)) {
            System.out.println(number + " is a perfect number.");
        } else {
            System.out.println(number + " is not a perfect number.");
        }
    }

    // Method to check if a number is a perfect number
    public static boolean isPerfectNumber(int number) {
        if (number < 1) {
            return false; // Perfect numbers are positive integers
        }

        int sum = 0;
        // Find all divisors and add them
        for (int i = 1; i <= number / 2; i++) {
            if (number % i == 0) {
                sum += i;
            }
        }

        // Check if the sum of divisors is equal to the number
        return sum == number;
    }
}
```



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

```
package com.example.main;

import java.util.Scanner;

public class Program {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number1 = sc.nextInt(); // 220

        System.out.print("Enter the number : ");
        int number2 = sc.nextInt(); // 284

        if (areAmicableNumbers(number1, number2)) {
            System.out.println(number1 + " and " + number2 + " are amicable numbers.");
        } else {
            System.out.println(number1 + " and " + number2 + " are not amicable numbers.");
        }
    }

    // Method to check if a pair of numbers are amicable
    public static boolean areAmicableNumbers(int num1, int num2) {
        return (sumOfProperDivisors(num1) == num2 && sumOfProperDivisors(num2) == num1);
    }

    // Method to calculate the sum of proper divisors of a number
    public static int sumOfProperDivisors(int num) {
        int sum = 0;
        for (int i = 1; i <= num / 2; i++) {
            if (num % i == 0) {
                sum += i;
            }
        }
        return sum;
    }
}
```

Problems @ Javadoc Declaration Console X

<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot

Enter the number : 1184

Enter the number : 1210

1184 and 1210 are amicable numbers.

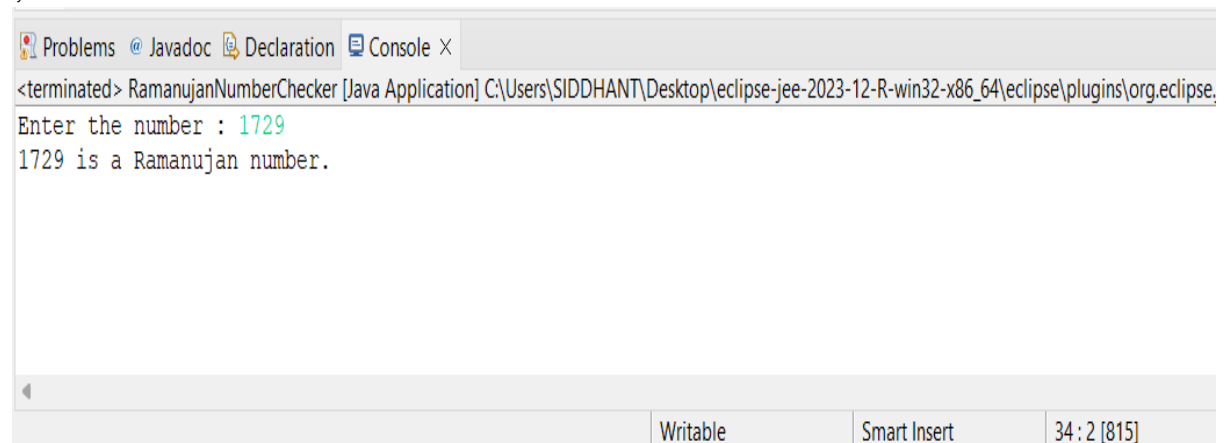
9. Write a Java Program to check whether the given number is Ramanujan's Number or NOT.

```
package com.example.main;

import java.util.Scanner;

public class RamanujanNumberChecker {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number = sc.nextInt();
        if (isRamanujanNumber(number)) {
            System.out.println(number + " is a Ramanujan
number.");
        }
        else {
            System.out.println(number + " is not a
Ramanujan number.");
        }
    }

    public static boolean isRamanujanNumber(int n) {
        int count = 0;
        int limit = (int) Math.cbrt(n);
        for (int i = 1; i <= limit; i++) {
            for (int j = i + 1; j <= limit; j++) {
                int sum = (int) (Math.pow(i, 3) + Math.pow(j, 3));
                if (sum == n) {
                    count++;
                    if (count == 2) {
                        return true;
                    }
                }
            }
        }
        return false;
    }
}
```



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The console output displays the program's execution: 'Enter the number : 1729' followed by '1729 is a Ramanujan number.' The IDE's status bar at the bottom indicates 'Writable', 'Smart Insert', and '34 : 2 [815]'.

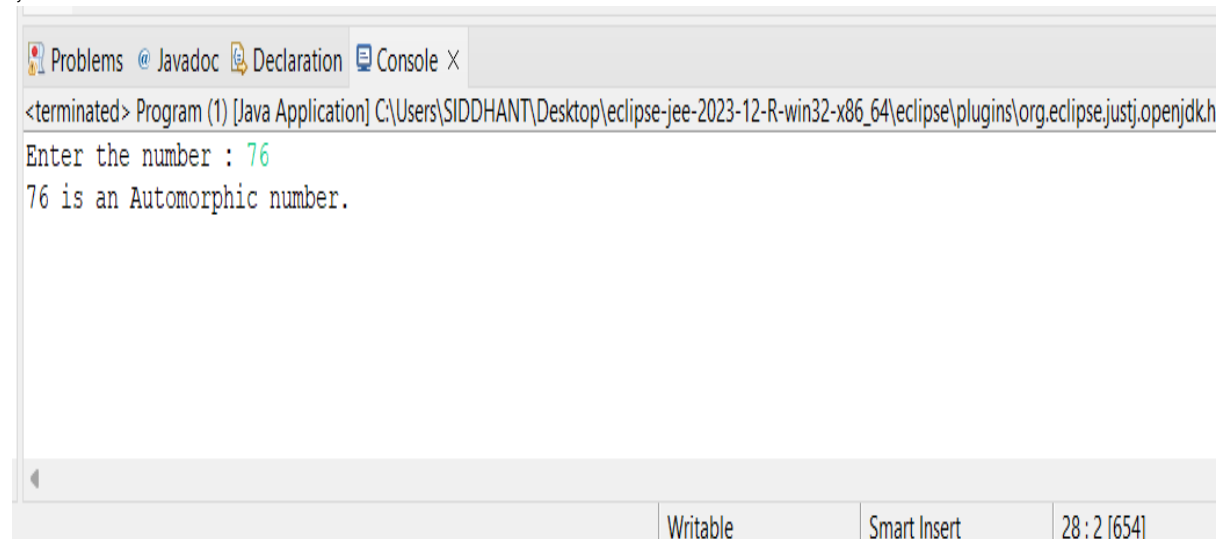
10. Write a Java Program check whether the given number is Automorphic Number or NOT.

```
package com.example.main;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number = sc.nextInt(); // 5,25,6,36,
        if (isAutomorphicNumber(number)) {
            System.out.println(number + " is an Automorphic
number.");
        } else {
            System.out.println(number + " is not an Automorphic
number.");
        }
    }

    public static boolean isAutomorphicNumber(int n) {
        int square = n * n;
        while (n > 0) {
            if (n % 10 != square % 10) {
                return false;
            }
            n /= 10;
            square /= 10;
        }
        return true;
    }
}
```



```
<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.h
Enter the number : 76
76 is an Automorphic number.
```

Writable Smart Insert 28 : 2 [654]