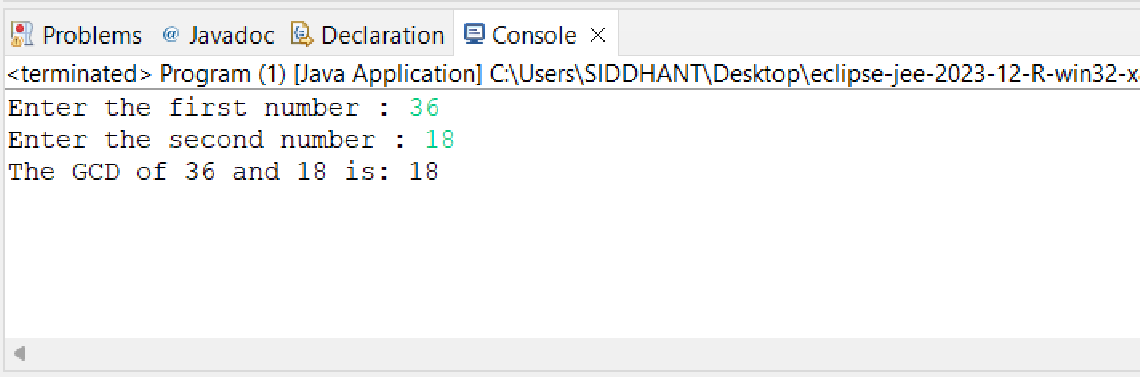
1. Write a Java Program to find GCD of two given numbers.

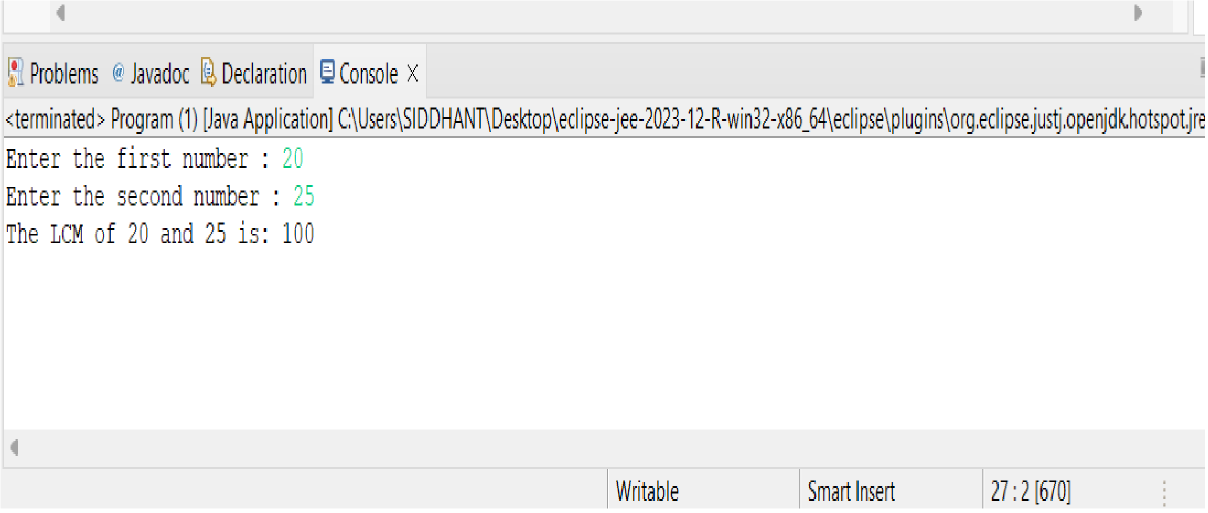
```
package com.example.main;

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);
            }
      }
}
```



```
<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
The GCD of 36 and 18 is: 18
```

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

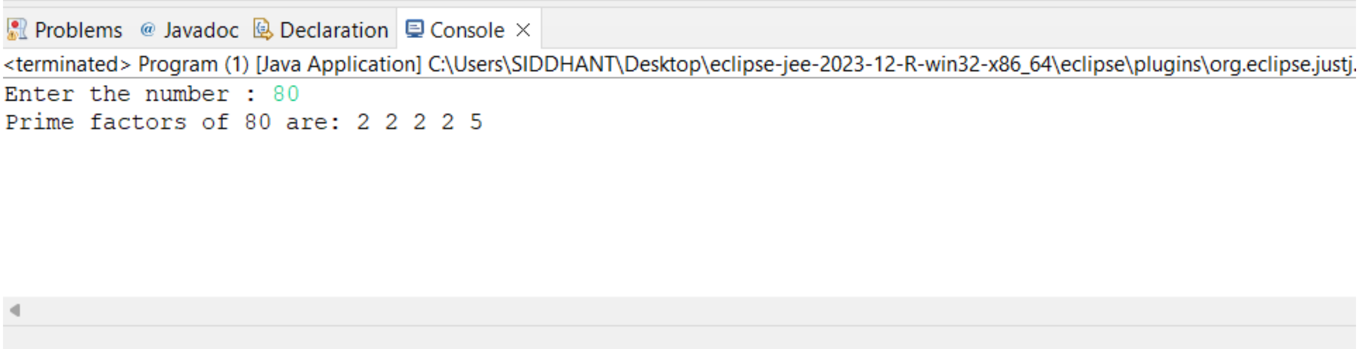
```
package com.example.main;

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);
		}
	}
}
```



```
<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
```

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

```java
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);
        }
    }
}
```

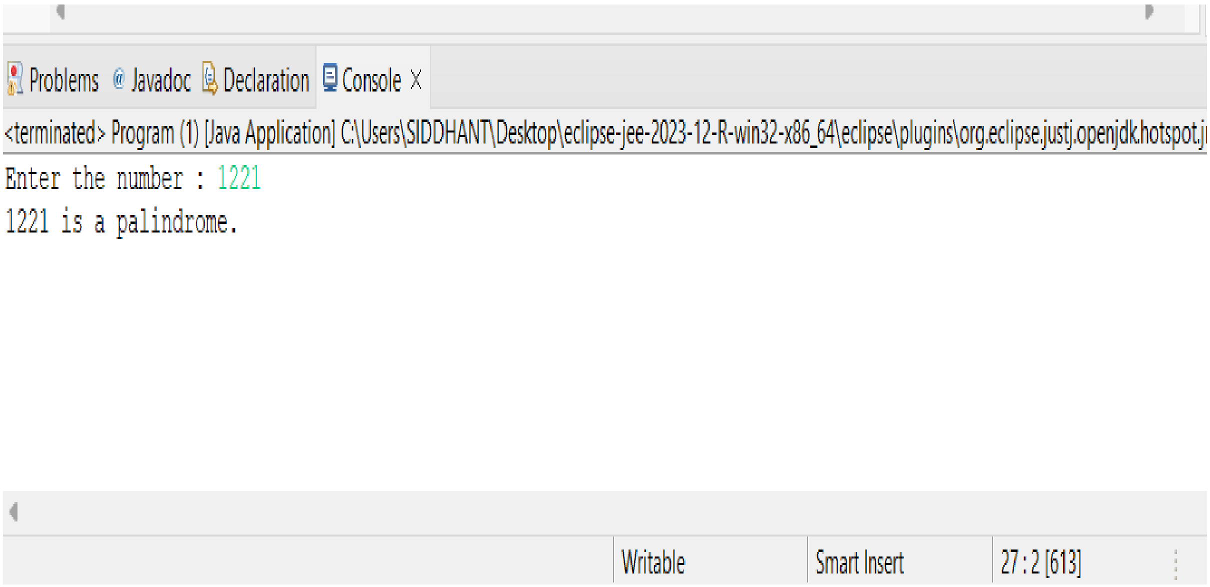


<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.

```
Enter the number : 80
Prime factors of 80 are: 2 2 2 2 5
```

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

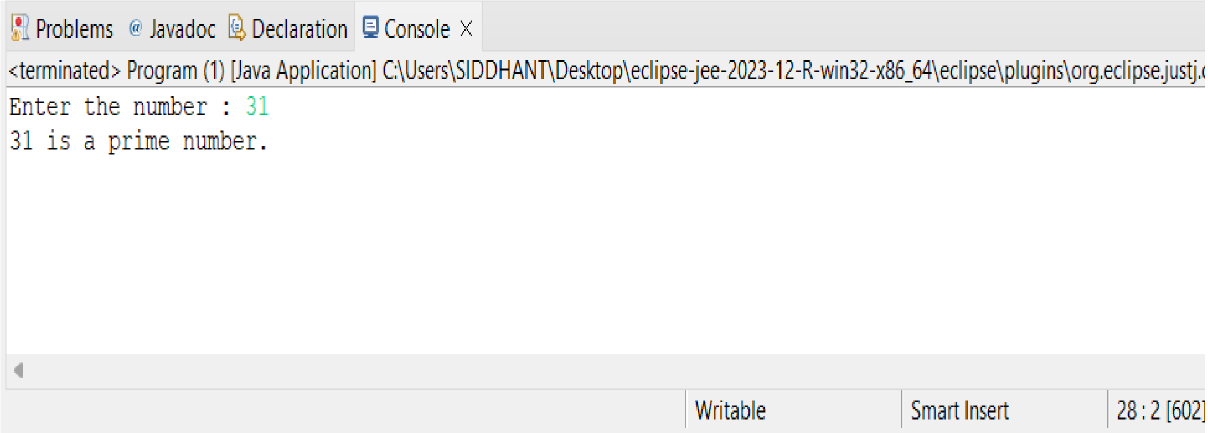
```java
public class Program {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.print("Enter the number : ");
		int number = sc.nextInt();
		if (isPalindrome(number)) {
			System.out.println(number + " is a palindrome.");
		} else {
			System.out.println(number + " 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;
	}
}
```



```
<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.
```

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

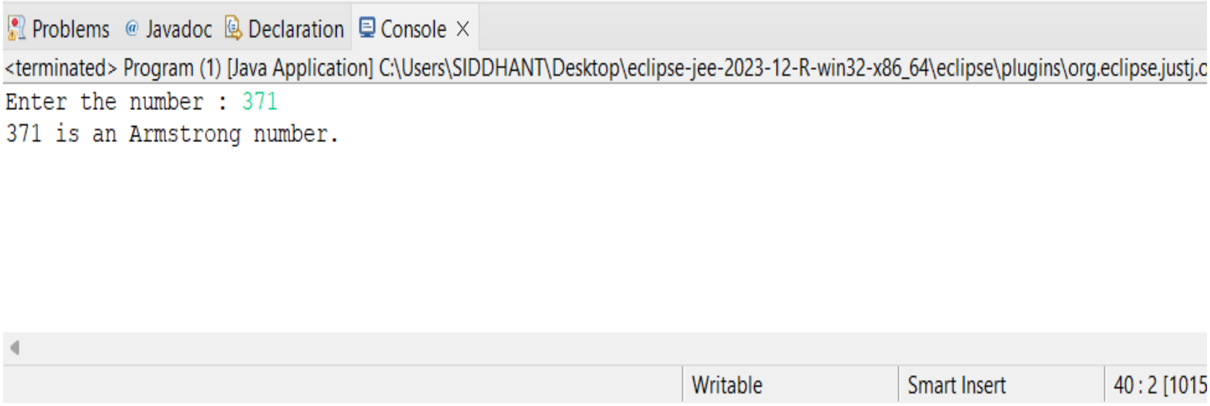
```java
package com.example.main;

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;
	}
}
```



```
<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.
```

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

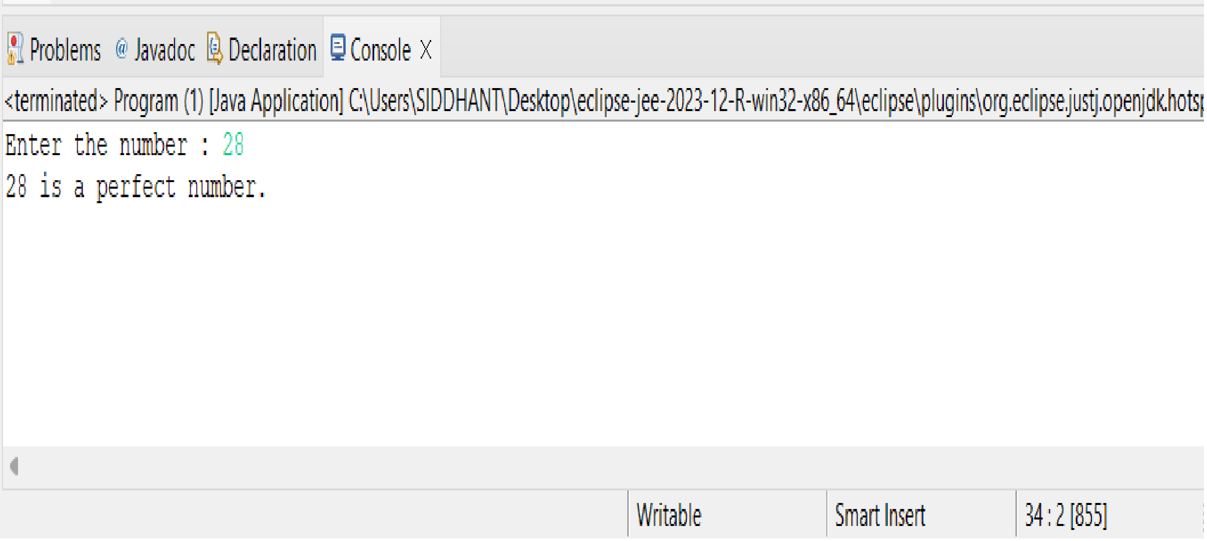
```java
package com.example.main;

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;
        }
    }
}
```



```
<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.
```

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

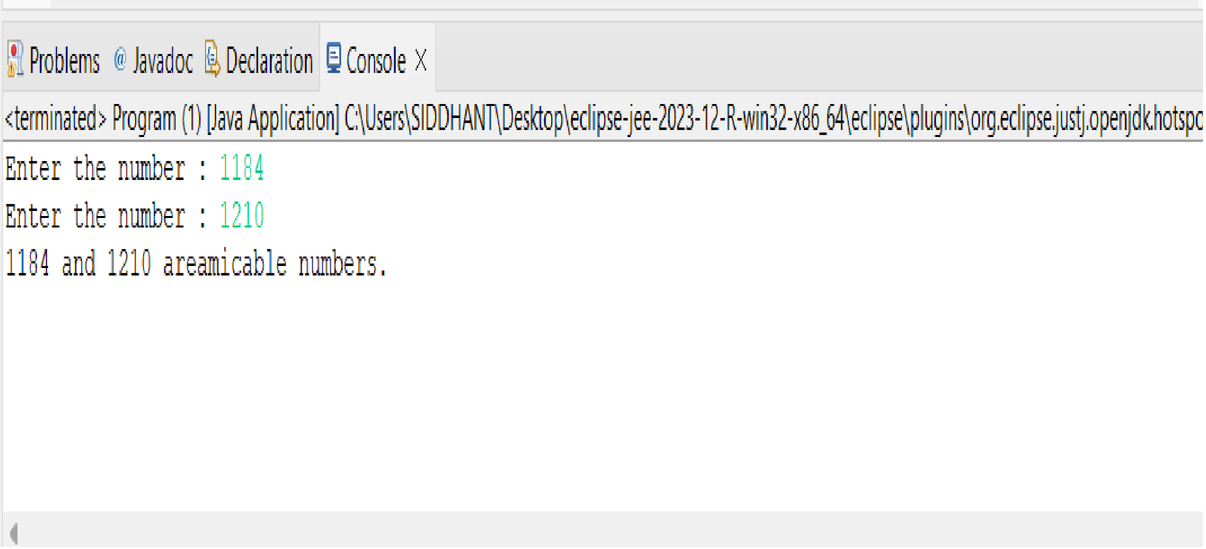
```java
package com.example.main;

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;
    }
}
```



```
<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotsp
Enter the number : 28
28 is a perfect number.
```

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

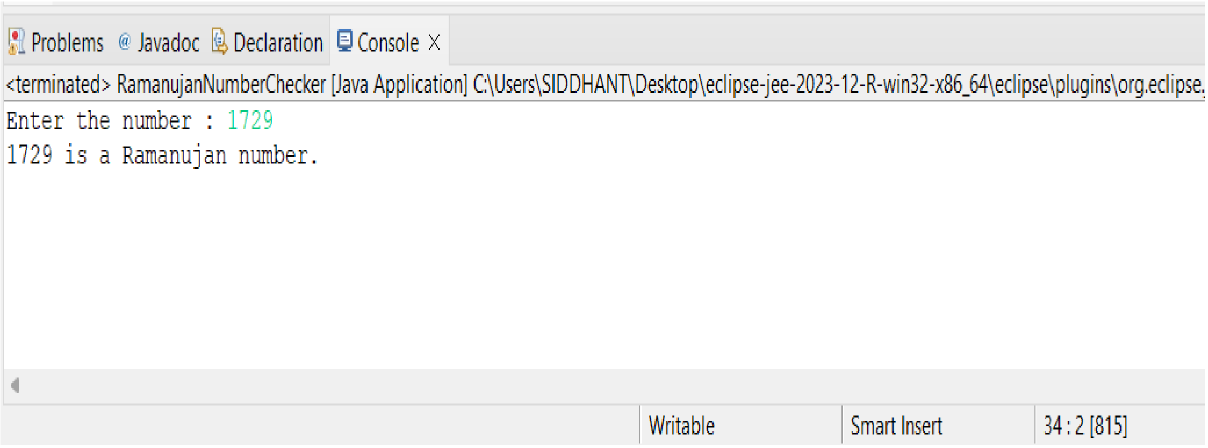
```java
package com.example.main;

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 + " 
areamicable 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;
    }
}
```



```
<terminated> Program (1) [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspc
Enter the number : 1184
Enter the number : 1210
1184 and 1210 areamicable numbers.
```

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

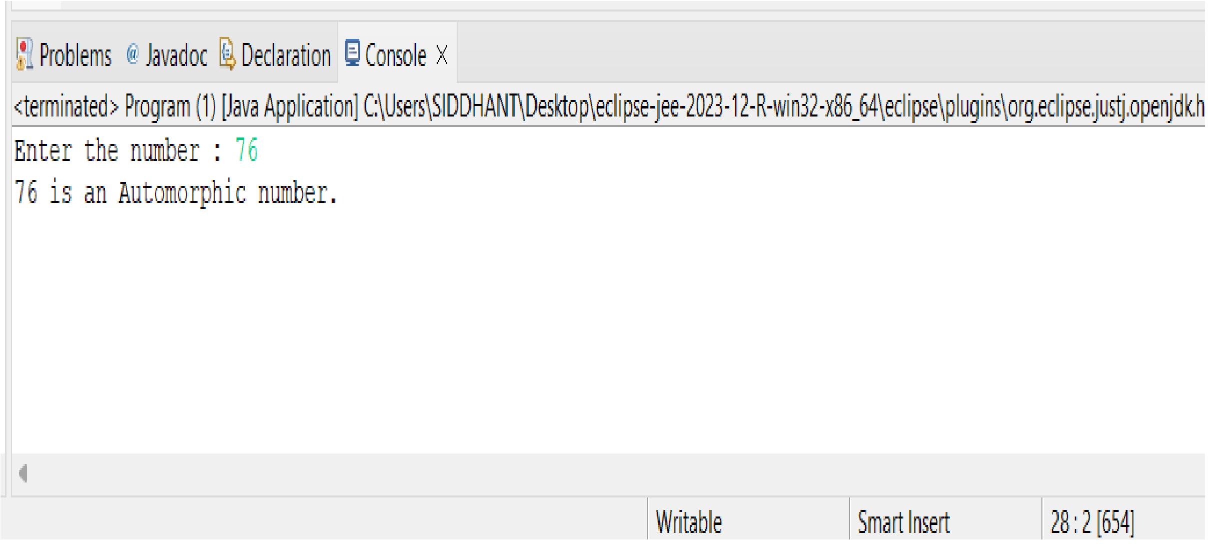
```java
package com.example.main;

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
Ramanujannumber.");
            }
    }

    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;
    }
}
```



```
<terminated> RamanujanNumberChecker [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.
Enter the number : 1729
1729 is a Ramanujan number.
```

10. Write a Java Program check whether the given number is Automorphic Numberor 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.
```