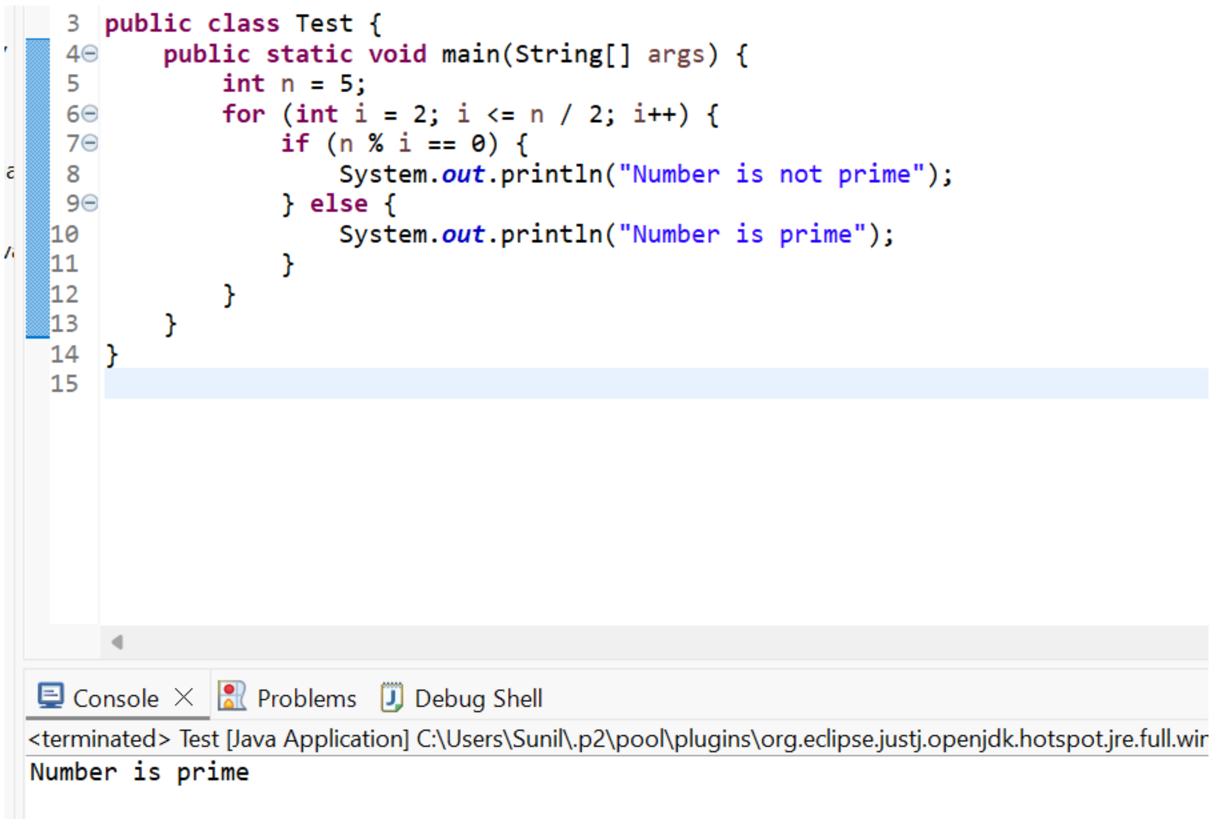
1. Java program to Find Odd or Even number

```
  import java.util.Scanner;

  public class OddEven {
        public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter any number: ");
        int number = scanner.nextInt();

        if (number % 2 == 0) {
            System.out.println(number + " is even.");
        } else {
            System.out.println(number + " is odd.");
        }
    }
}
```

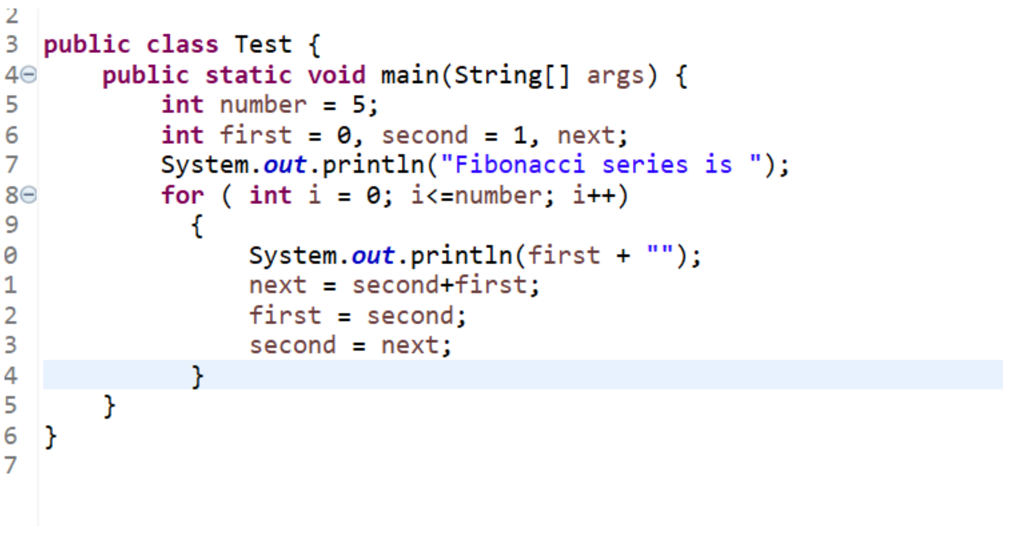
2. Java program to find Prime number



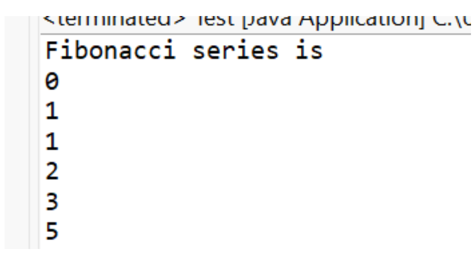
```
public class Test {
    public static void main(String[] args) {
        int n = 5;
        for (int i = 2; i <= n / 2; i++) {
            if (n % i == 0) {
                System.out.println("Number is not prime");
            } else {
                System.out.println("Number is prime");
            }
        }
    }
}
```

```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wir
Number is prime
```

3. Java program to find Fibonacci series up to a given number range

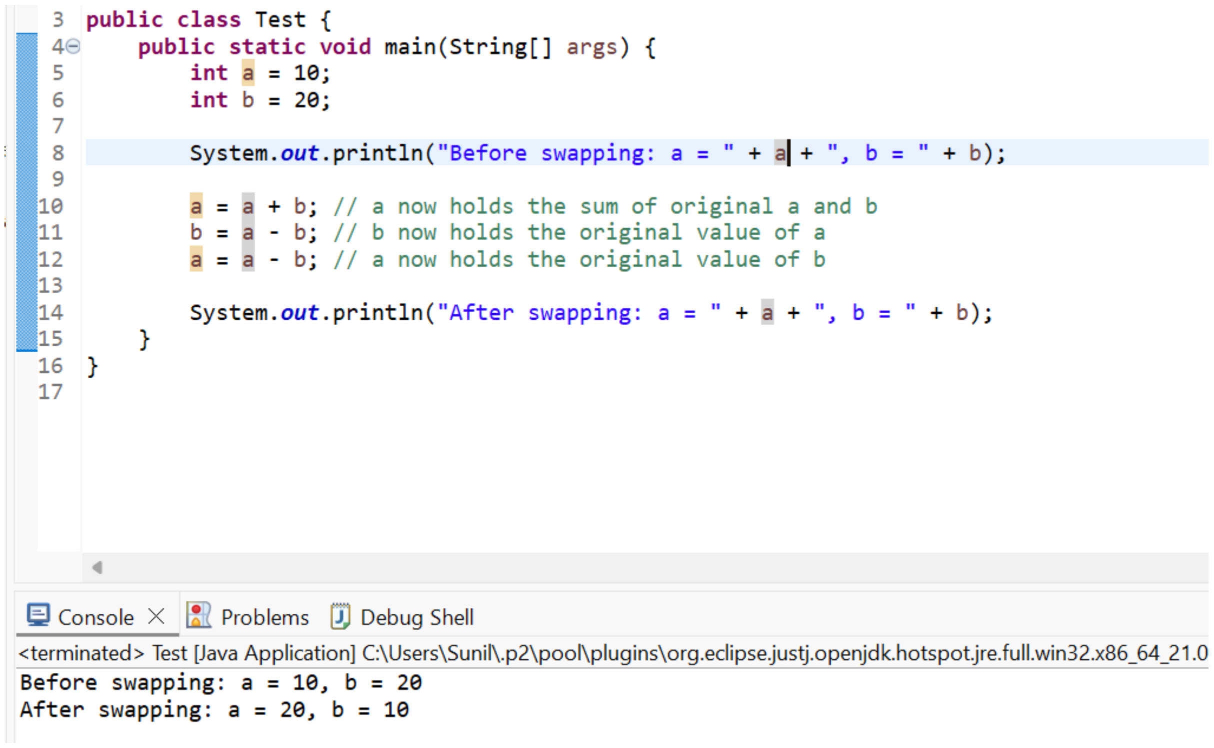


```java
public class Test {
    public static void main(String[] args) {
        int number = 5;
        int first = 0, second = 1, next;
        System.out.println("Fibonacci series is ");
        for ( int i = 0; i<=number; i++)
          {
              System.out.println(first + "");
              next = second+first;
              first = second;
              second = next;
          }
    }
}
```



```
Fibonacci series is
0
1
1
2
3
5
```

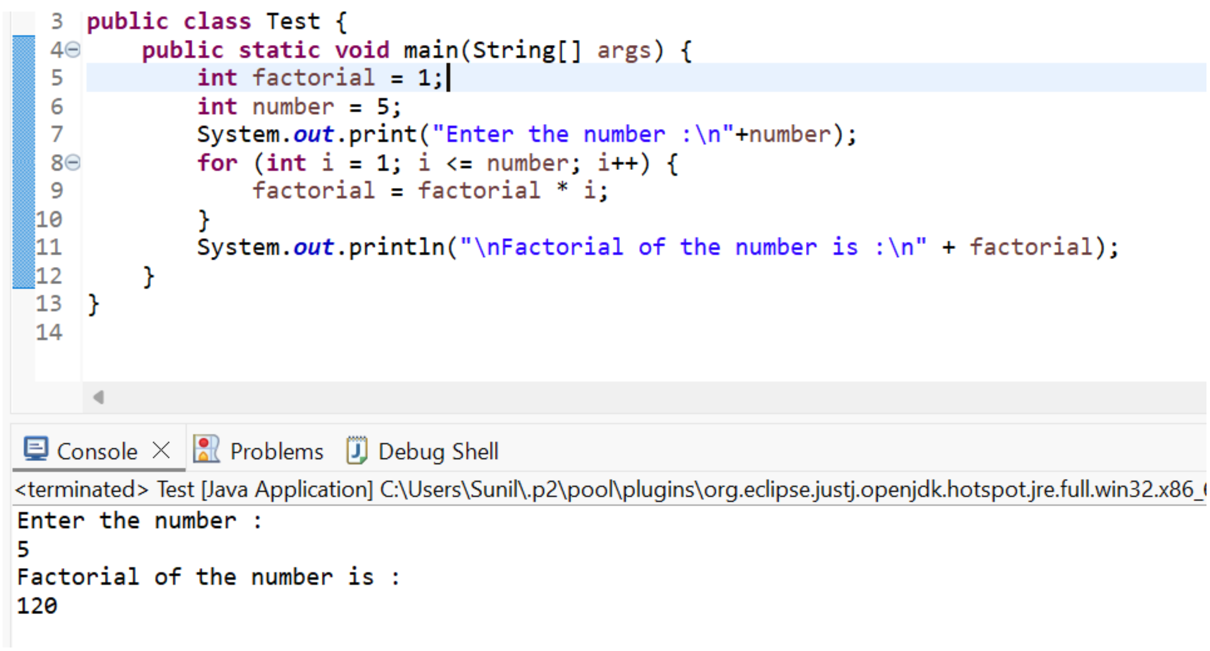
4. Java program to swap two numbers without using third variable



```java
public class Test {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;

        System.out.println("Before swapping: a = " + a + ", b = " + b);

        a = a + b; // a now holds the sum of original a and b
        b = a - b; // b now holds the original value of a
        a = a - b; // a now holds the original value of b

        System.out.println("After swapping: a = " + a + ", b = " + b);
    }
}
```

```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0
Before swapping: a = 10, b = 20
After swapping: a = 20, b = 10
```

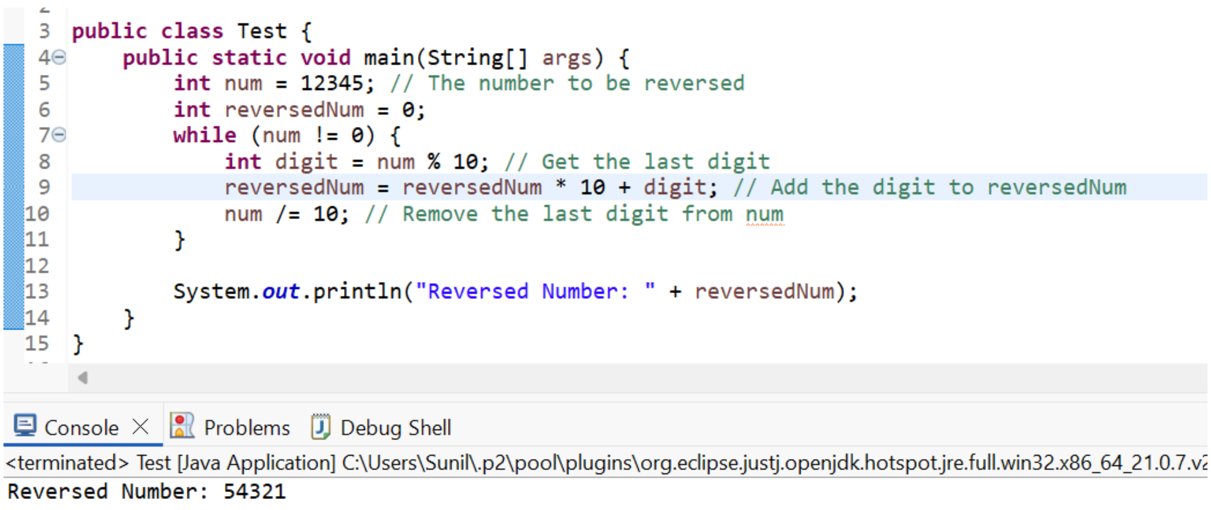
5. Java program to Find Factorial on given Number



```
public class Test {
    public static void main(String[] args) {
        int factorial = 1;
        int number = 5;
        System.out.print("Enter the number :\n"+number);
        for (int i = 1; i <= number; i++) {
            factorial = factorial * i;
        }
        System.out.println("\nFactorial of the number is :\n" + factorial);
    }
}
```

```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_
Enter the number :
5
Factorial of the number is :
120
```

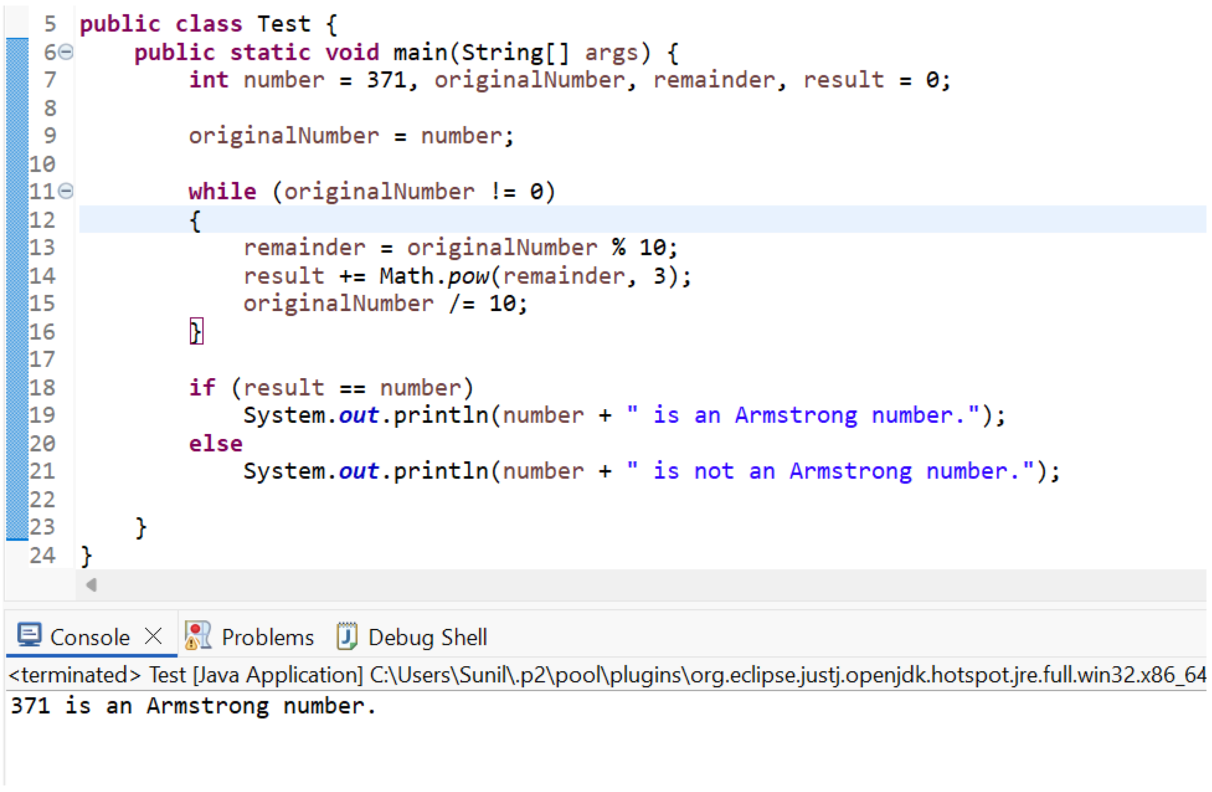
6. Java program to Reverse Number



```
public class Test {
    public static void main(String[] args) {
        int num = 12345; // The number to be reversed
        int reversedNum = 0;
        while (num != 0) {
            int digit = num % 10; // Get the last digit
            reversedNum = reversedNum * 10 + digit; // Add the digit to reversedNum
            num /= 10; // Remove the last digit from num
        }

        System.out.println("Reversed Number: " + reversedNum);
    }
}
```

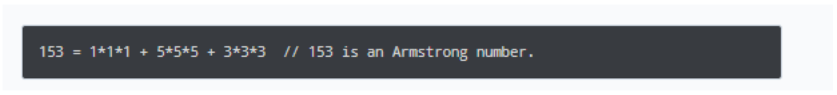
```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v
Reversed Number: 54321
```

7. Java program to find Armstrong Number



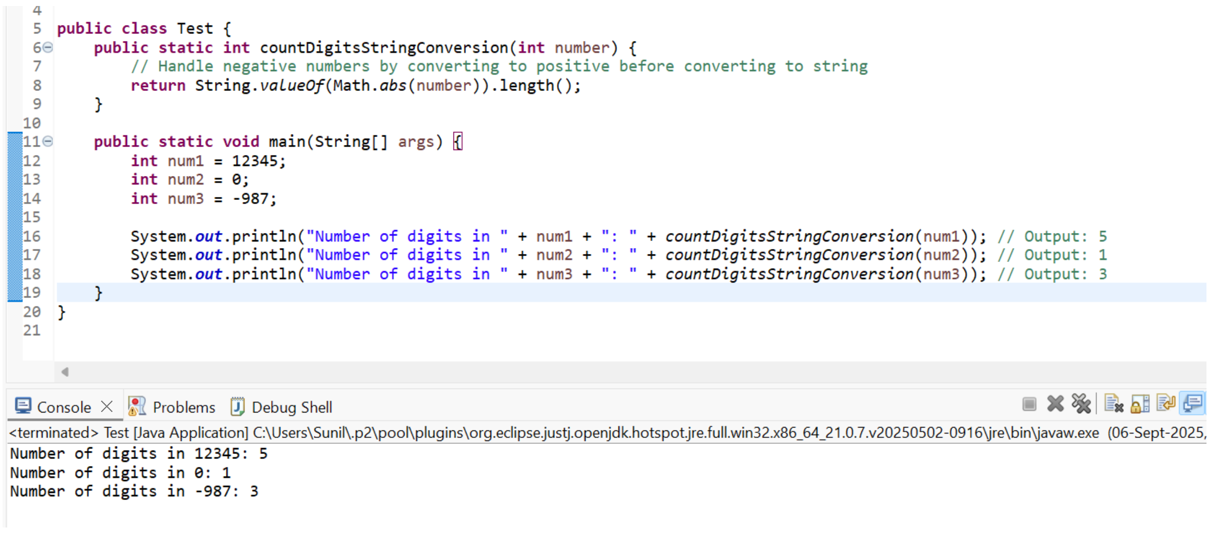
```java
public class Test {
    public static void main(String[] args) {
        int number = 371, originalNumber, remainder, result = 0;

        originalNumber = number;

        while (originalNumber != 0)
        {
            remainder = originalNumber % 10;
            result += Math.pow(remainder, 3);
            originalNumber /= 10;
        }

        if (result == number)
            System.out.println(number + " is an Armstrong number.");
        else
            System.out.println(number + " is not an Armstrong number.");

    }
}
```

```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
371 is an Armstrong number.
```



```
153 = 1*1*1 + 5*5*5 + 3*3*3  // 153 is an Armstrong number.
```

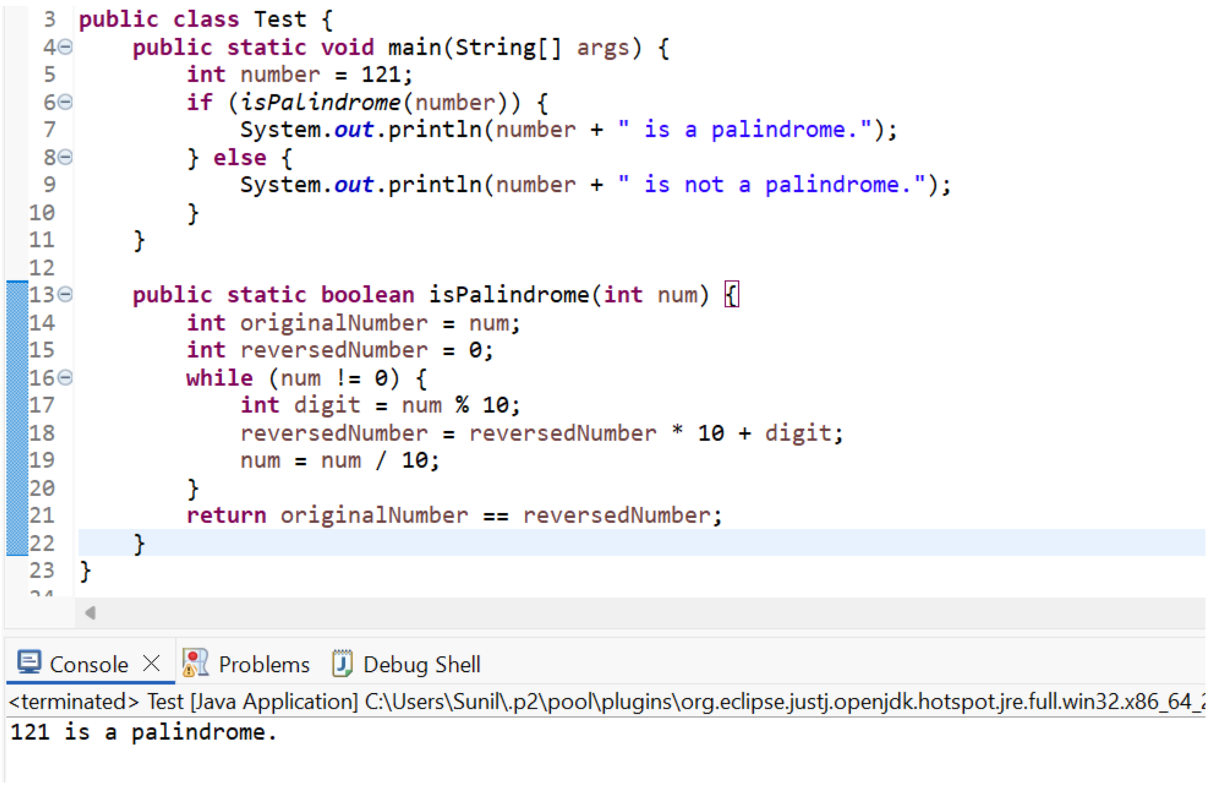
8. Java program to find number of digits in given number



```java
public class Test {
    public static int countDigitsStringConversion(int number) {
        // Handle negative numbers by converting to positive before converting to string
        return String.valueOf(Math.abs(number)).length();
    }

    public static void main(String[] args) {
        int num1 = 12345;
        int num2 = 0;
        int num3 = -987;

        System.out.println("Number of digits in " + num1 + ": " + countDigitsStringConversion(num1)); // Output: 5
        System.out.println("Number of digits in " + num2 + ": " + countDigitsStringConversion(num2)); // Output: 1
        System.out.println("Number of digits in " + num3 + ": " + countDigitsStringConversion(num3)); // Output: 3
    }
}
```

```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe  (06-Sept-2025,
Number of digits in 12345: 5
Number of digits in 0: 1
Number of digits in -987: 3
```

9. Java program to find Palindrome number

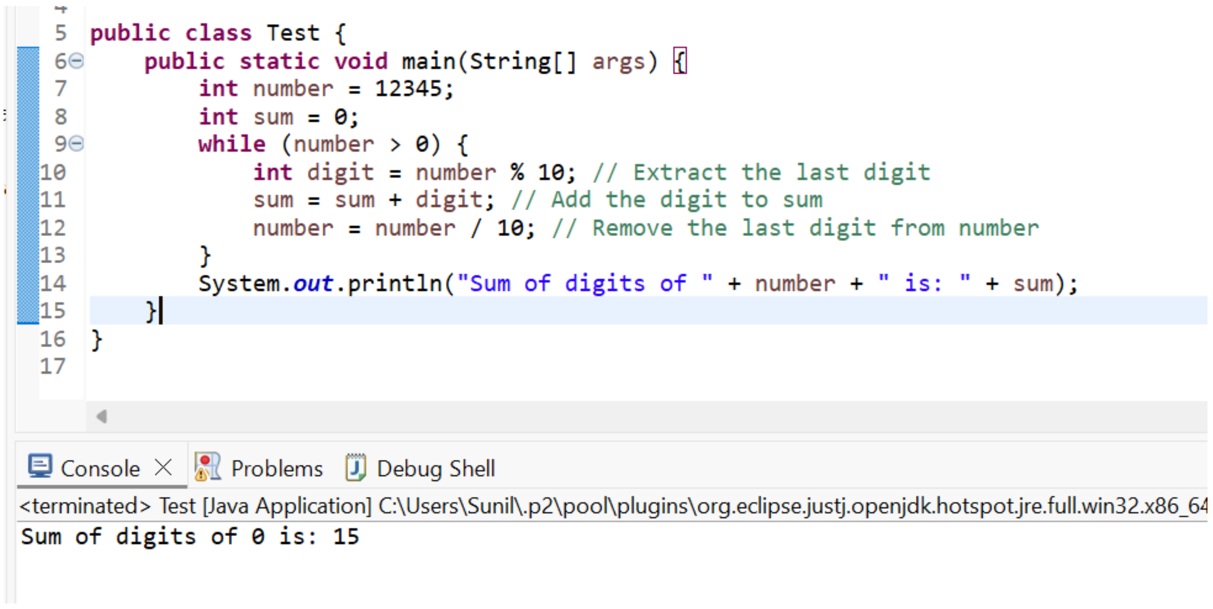


```
public class Test {
    public static void main(String[] args) {
        int number = 121;
        if (isPalindrome(number)) {
            System.out.println(number + " is a palindrome.");
        } else {
            System.out.println(number + " is not a palindrome.");
        }
    }

    public static boolean isPalindrome(int num) {
        int originalNumber = num;
        int reversedNumber = 0;
        while (num != 0) {
            int digit = num % 10;
            reversedNumber = reversedNumber * 10 + digit;
            num = num / 10;
        }
        return originalNumber == reversedNumber;
    }
}
```

Console Problems Debug Shell

```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_
121 is a palindrome.
```

10. Java program to calculate the sum of digits of a number



```
public class Test {
    public static void main(String[] args) {
        int number = 12345;
        int sum = 0;
        while (number > 0) {
            int digit = number % 10; // Extract the last digit
            sum = sum + digit; // Add the digit to sum
            number = number / 10; // Remove the last digit from number
        }
        System.out.println("Sum of digits of " + number + " is: " + sum);
    }
}
```

Console Problems Debug Shell

```
<terminated> Test [Java Application] C:\Users\Sunil\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
Sum of digits of 0 is: 15
```