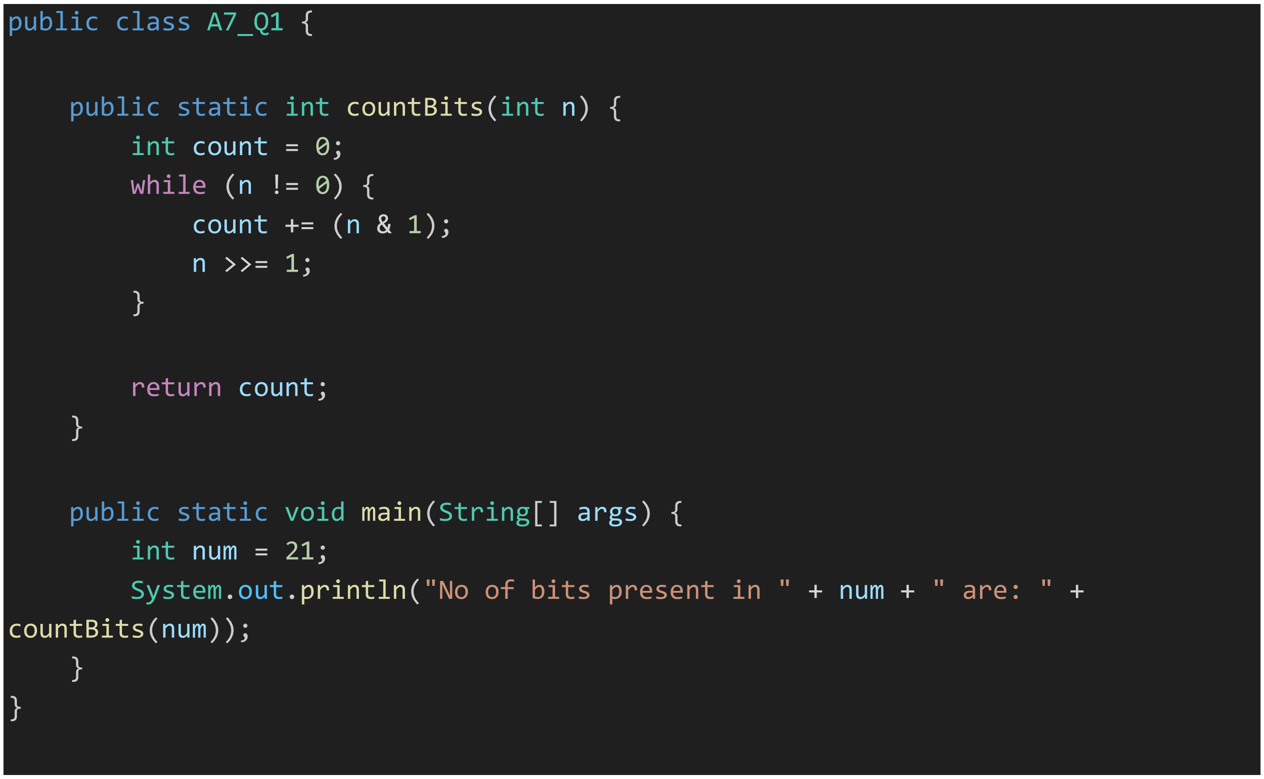
Assignment –7(csw-1)

Anupam chandra

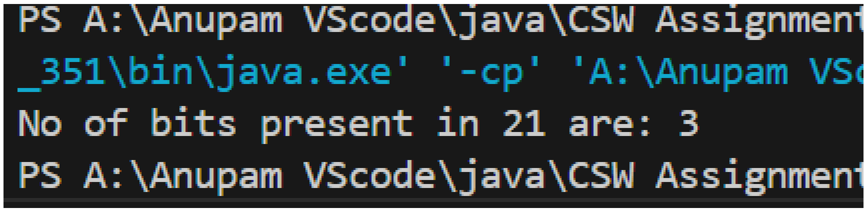
2341003015

23412G1

Q1. Write a Java program to count the number of bits that are set to 1 in an integer.

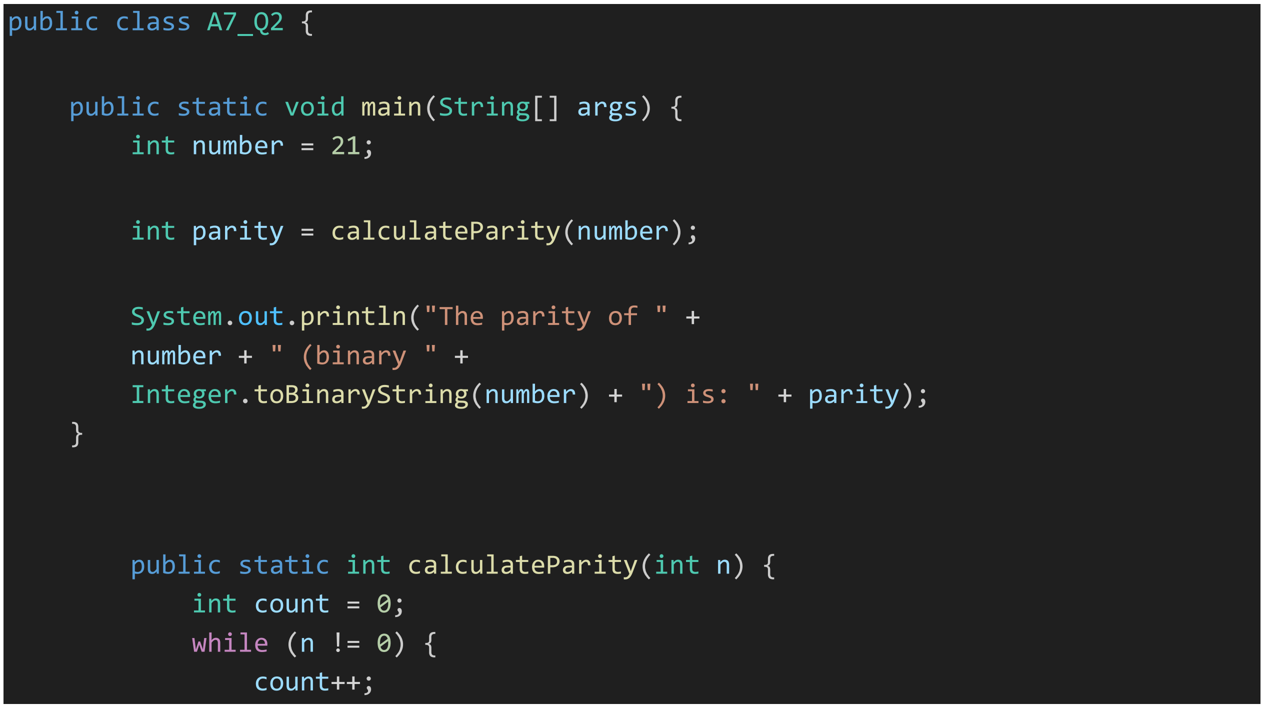


```
public class A7_Q1 {

    public static int countBits(int n) {
        int count = 0;
        while (n != 0) {
            count += (n & 1);
            n >>= 1;
        }

        return count;
    }

    public static void main(String[] args) {
        int num = 21;
        System.out.println("No of bits present in " + num + " are: " +
countBits(num));
    }
}
```

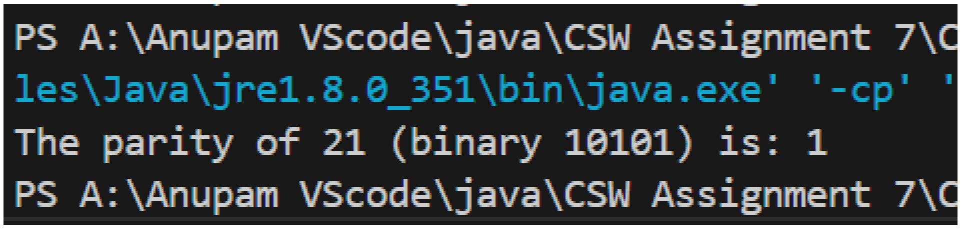


```
PS A:\Anupam VScode\java\CSW Assignment
_351\bin\java.exe' '-cp' 'A:\Anupam VSc
No of bits present in 21 are: 3
PS A:\Anupam VScode\java\CSW Assignment
```

Q2. The parity of a binary word is 1 if the number of 1s in the word is odd; otherwise, it is 0. Write a Java program to count the parity of an integer number.



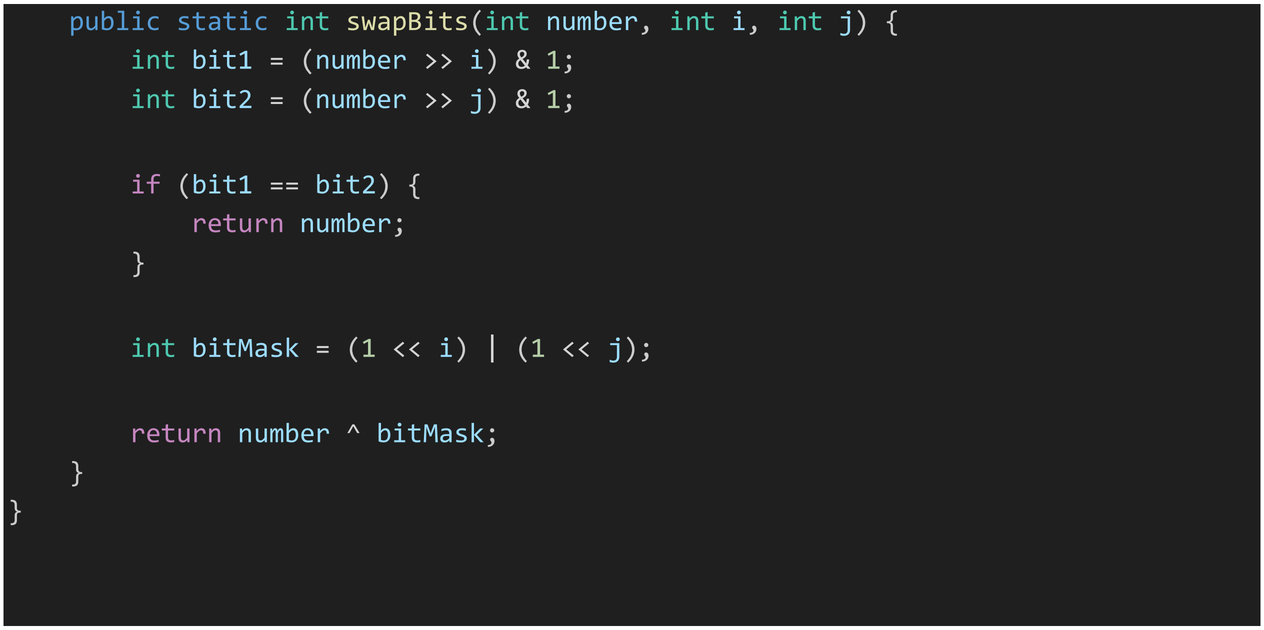
```
public class A7_Q2 {

    public static void main(String[] args) {
        int number = 21;

        int parity = calculateParity(number);

        System.out.println("The parity of " +
        number + " (binary " +
        Integer.toBinaryString(number) + ") is: " + parity);
    }


        public static int calculateParity(int n) {
            int count = 0;
            while (n != 0) {
                count++;
                n &= (n - 1);
            }

            return count % 2 == 1 ? 1 : 0;
        }

}
```



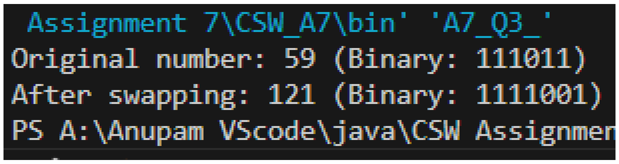
```
PS A:\Anupam VScode\java\CSW Assignment 7\C
les\Java\jre1.8.0_351\bin\java.exe' '-cp' '
The parity of 21 (binary 10101) is: 1
PS A:\Anupam VScode\java\CSW Assignment 7\C
```

Q3. Write a program to swap the ith bit with jth bit of a number.

```
public class A7_Q3_ {

    public static void main(String[] args) {
        int number = 59;
        int i = 1;
        int j = 6;

        System.out.println("Original number: " + number + " (Binary: " +
Integer.toBinaryString(number) + ")");
        int result = swapBits(number, i, j);
        System.out.println("After swapping: " + result + " (Binary: " +
Integer.toBinaryString(result) + ")");
    }
```

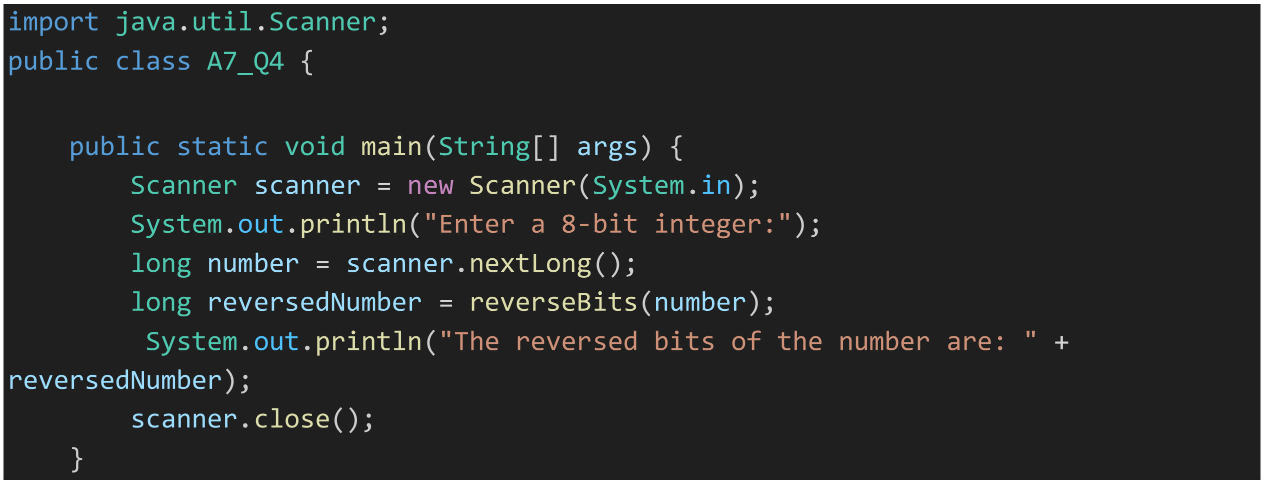


```
    public static int swapBits(int number, int i, int j) {
        int bit1 = (number >> i) & 1;
        int bit2 = (number >> j) & 1;

        if (bit1 == bit2) {
            return number;
        }

        int bitMask = (1 << i) | (1 << j);

        return number ^ bitMask;
    }
}
```

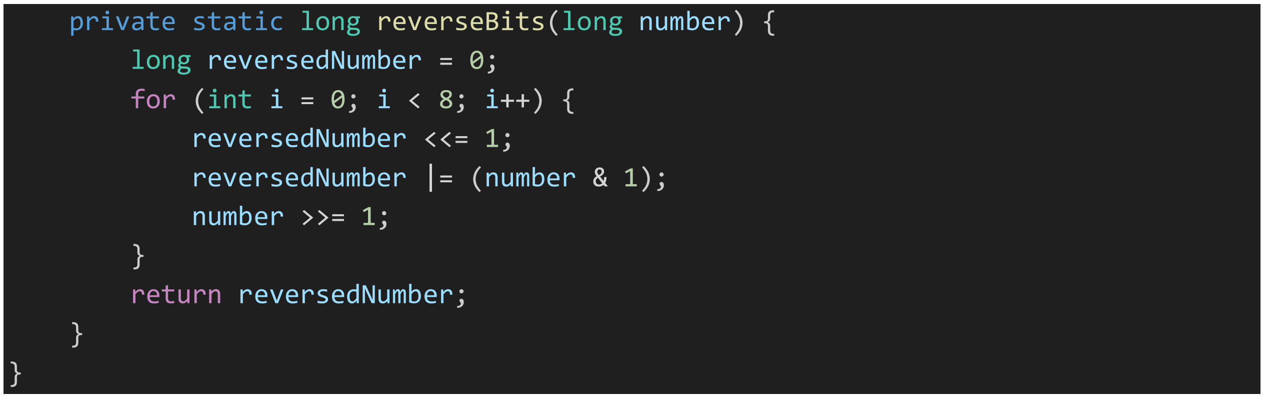


```
 Assignment 7\CSW_A7\bin' 'A7_Q3_'
Original number: 59 (Binary: 111011)
After swapping: 121 (Binary: 1111001)
PS A:\Anupam VScode\java\CSW Assignmen
```

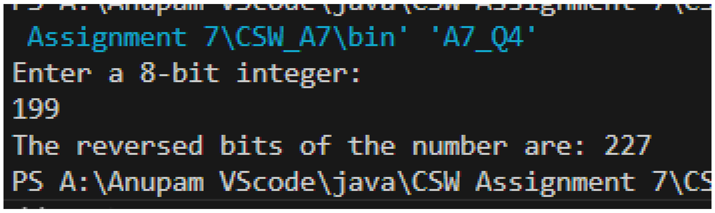
Q4. Write a program that takes a 64-bit word and returns the 64-bit word consisting of the bits of the input word in reverse order. For example, if the input is alternating 1s and 0s, i.e., (1010...10), the output should be alternating 0s and 1s, i.e.,(0101...01).



```
import java.util.Scanner;
public class A7_Q4 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a 8-bit integer:");
        long number = scanner.nextLong();
        long reversedNumber = reverseBits(number);
         System.out.println("The reversed bits of the number are: " +
reversedNumber);
        scanner.close();
    }
```

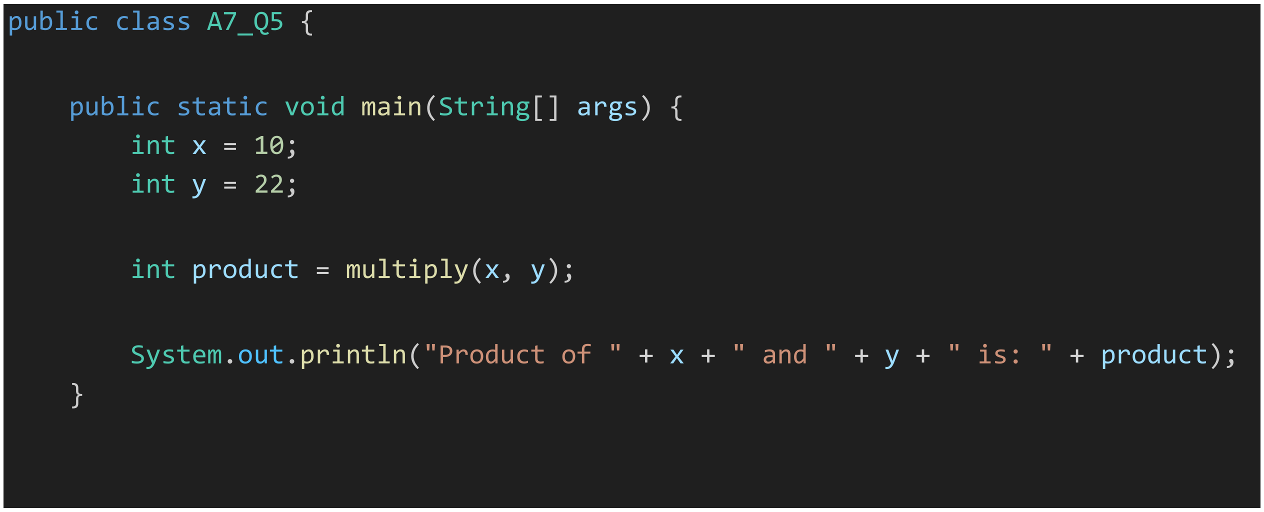


```
    private static long reverseBits(long number) {
        long reversedNumber = 0;
        for (int i = 0; i < 8; i++) {
            reversedNumber <<= 1;
            reversedNumber |= (number & 1);
            number >>= 1;
        }
        return reversedNumber;
    }
}
```

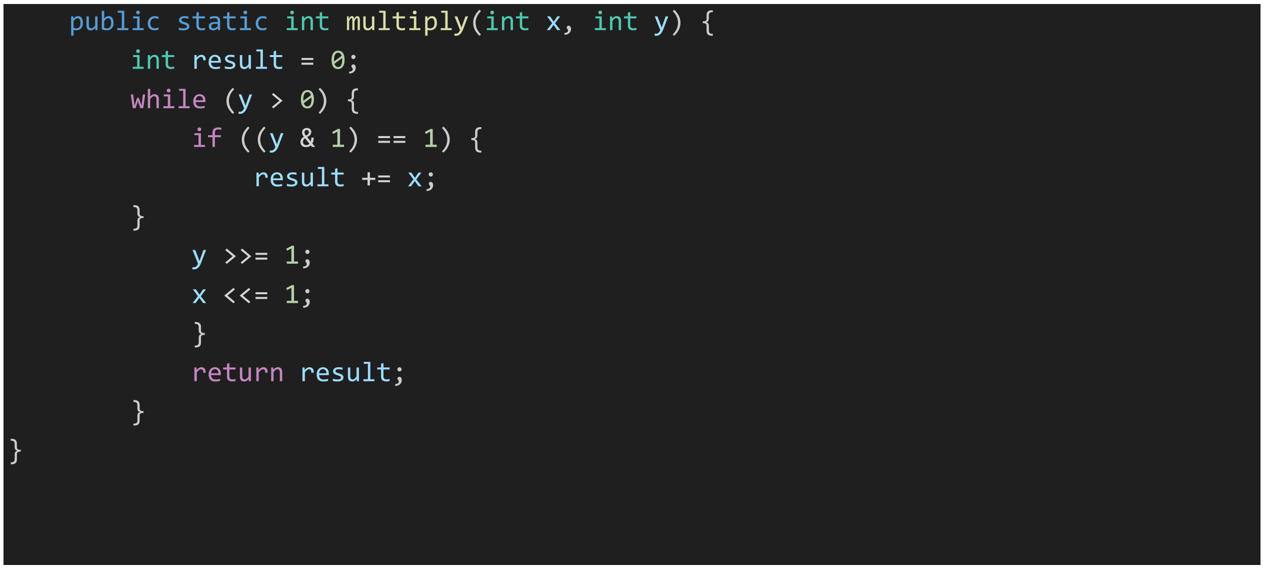


```
PS A:\Anupam VScode\java\CSW Assignment 7\CS
 Assignment 7\CSW_A7\bin' 'A7_Q4'
Enter a 8-bit integer:
199
The reversed bits of the number are: 227
PS A:\Anupam VScode\java\CSW Assignment 7\CS
```

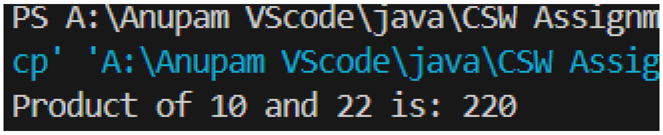
Q5. Write a java program to copmute x×y without arithmetic operators.



```
public class A7_Q5 {

    public static void main(String[] args) {
        int x = 10;
        int y = 22;

        int product = multiply(x, y);

        System.out.println("Product of " + x + " and " + y + " is: " + product);
    }
```

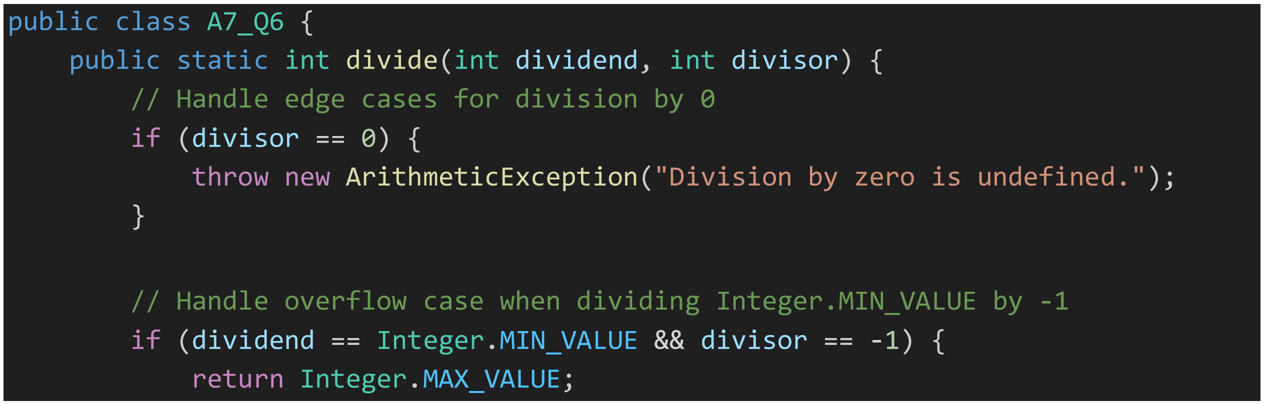


```
    public static int multiply(int x, int y) {
        int result = 0;
        while (y > 0) {
            if ((y & 1) == 1) {
                result += x;
        }
            y >>= 1;
            x <<= 1;
            }
            return result;
        }
}
```

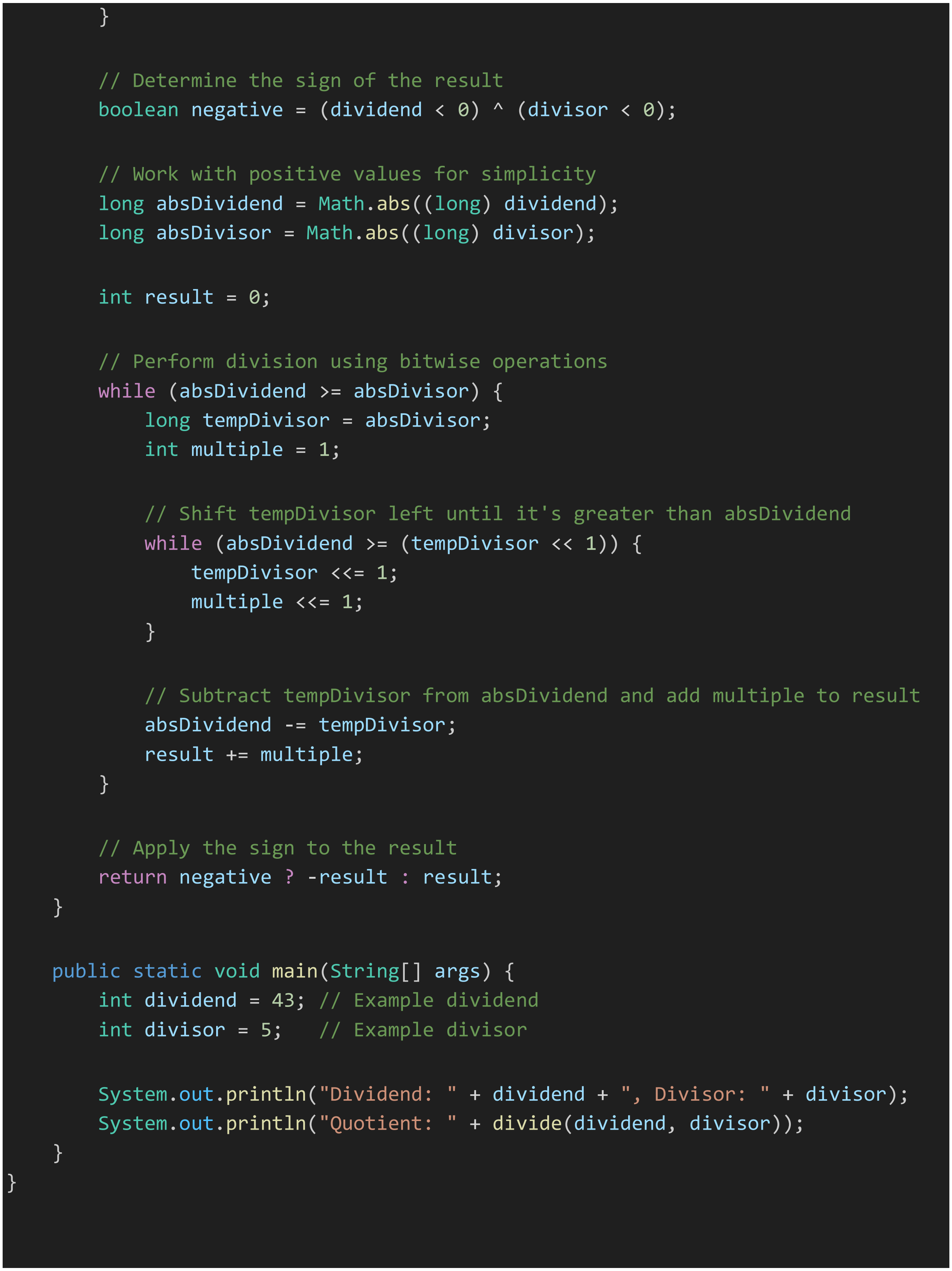


```
PS A:\Anupam VScode\java\CSW Assignm
cp' 'A:\Anupam VScode\java\CSW Assig
Product of 10 and 22 is: 220
```

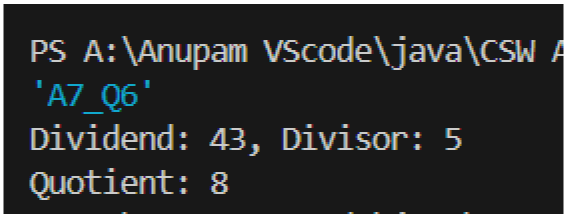
Q6. Write a java program to copmute x/y without arithmetic operators.



```
public class A7_Q6 {
    public static int divide(int dividend, int divisor) {
        // Handle edge cases for division by 0
        if (divisor == 0) {
            throw new ArithmeticException("Division by zero is undefined.");
        }

        // Handle overflow case when dividing Integer.MIN_VALUE by -1
        if (dividend == Integer.MIN_VALUE && divisor == -1) {
            return Integer.MAX_VALUE;
```

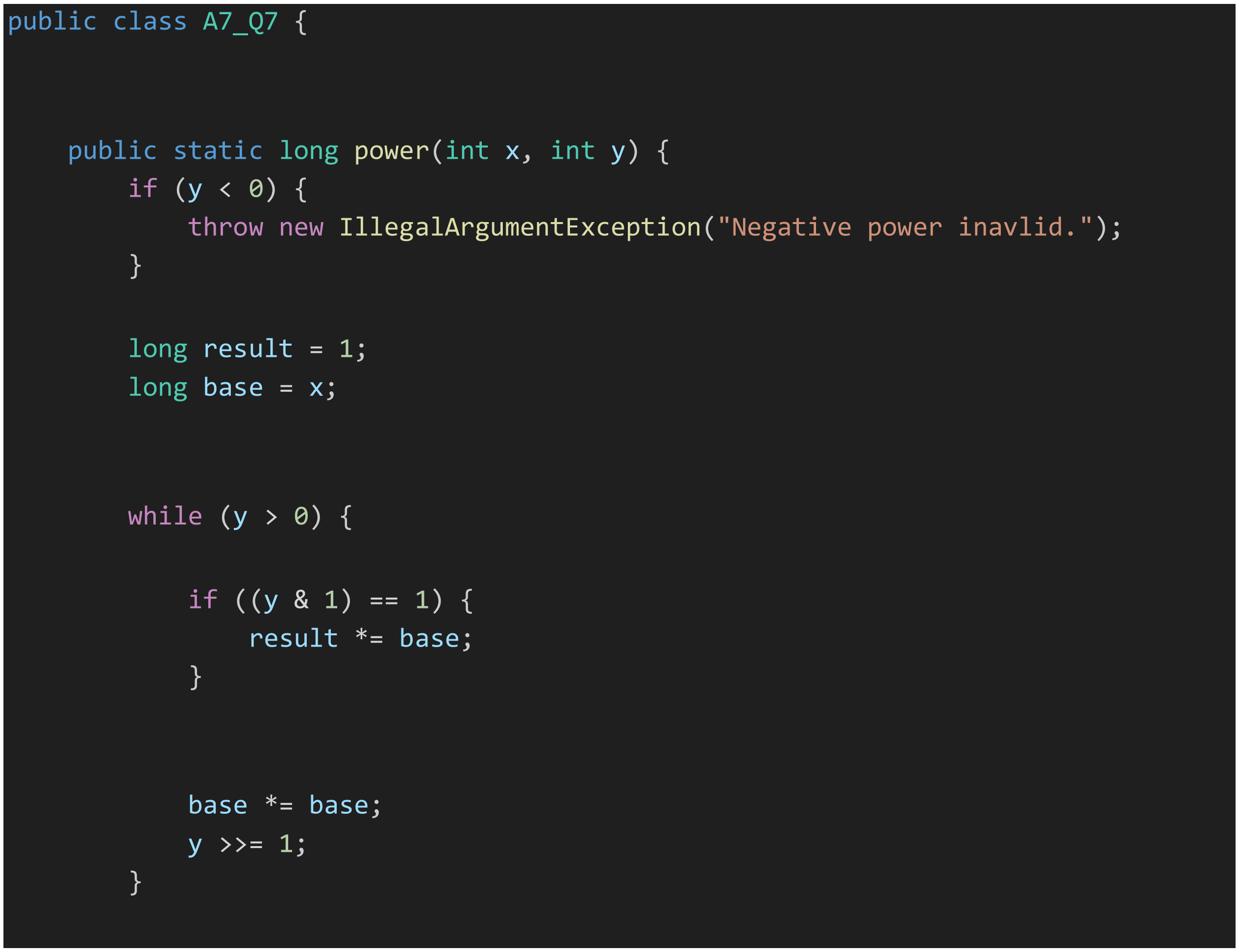


```
        }

        // Determine the sign of the result
        boolean negative = (dividend < 0) ^ (divisor < 0);

        // Work with positive values for simplicity
        long absDividend = Math.abs((long) dividend);
        long absDivisor = Math.abs((long) divisor);

        int result = 0;

        // Perform division using bitwise operations
        while (absDividend >= absDivisor) {
            long tempDivisor = absDivisor;
            int multiple = 1;

            // Shift tempDivisor left until it's greater than absDividend
            while (absDividend >= (tempDivisor << 1)) {
                tempDivisor <<= 1;
                multiple <<= 1;
            }

            // Subtract tempDivisor from absDividend and add multiple to result
            absDividend -= tempDivisor;
            result += multiple;
        }

        // Apply the sign to the result
        return negative ? -result : result;
    }

    public static void main(String[] args) {
        int dividend = 43; // Example dividend
        int divisor = 5;   // Example divisor

        System.out.println("Dividend: " + dividend + ", Divisor: " + divisor);
        System.out.println("Quotient: " + divide(dividend, divisor));
    }
}
```

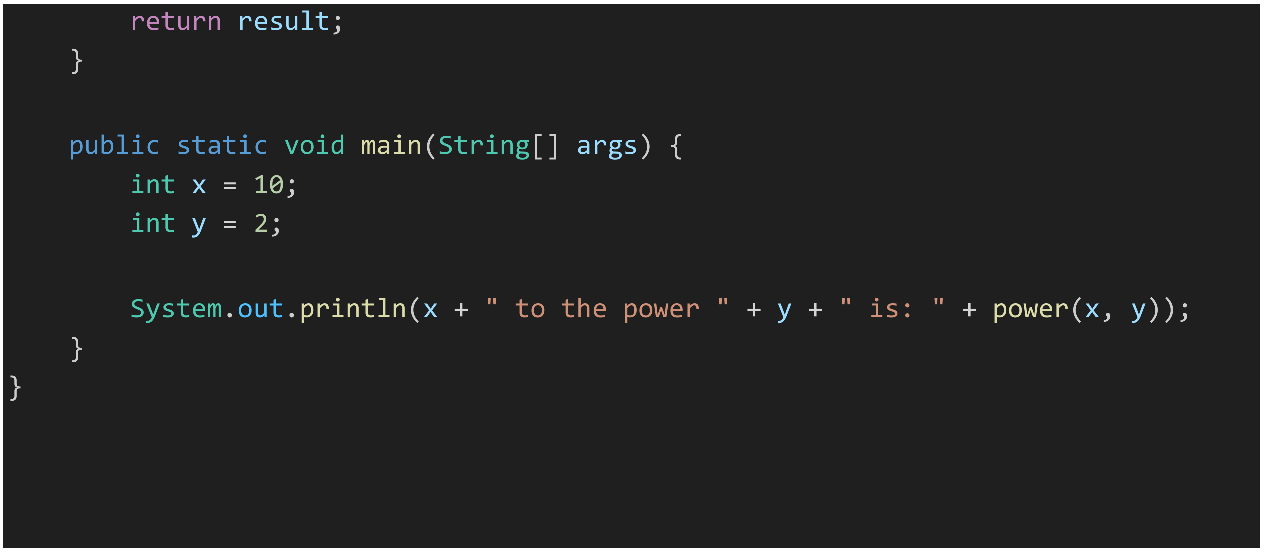


```
PS A:\Anupam VScode\java\CSW A
'A7_Q6'
Dividend: 43, Divisor: 5
Quotient: 8
```

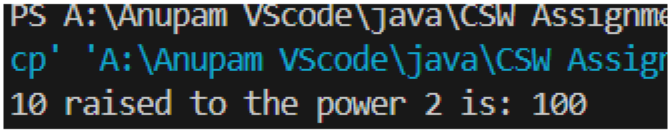
Q7. Write a program to find xy



```java
public class A7_Q7 {


    public static long power(int x, int y) {
        if (y < 0) {
            throw new IllegalArgumentException("Negative power inavlid.");
        }

        long result = 1;
        long base = x;


        while (y > 0) {

            if ((y & 1) == 1) {
                result *= base;
            }


            base *= base;
            y >>= 1;
        }
```

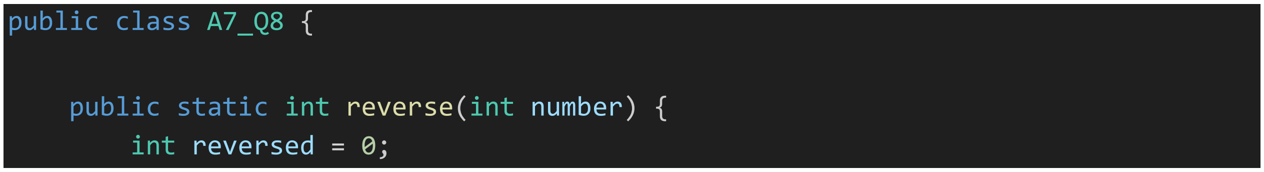


```java
        return result;
    }

    public static void main(String[] args) {
        int x = 10;
        int y = 2;

        System.out.println(x + " to the power " + y + " is: " + power(x, y));
    }
}
```

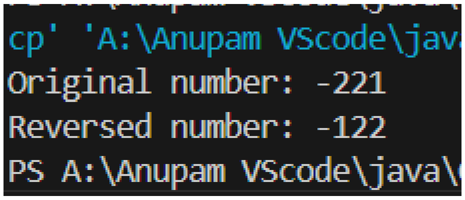


```
PS A:\Anupam VScode\java\CSW Assignme
cp' 'A:\Anupam VScode\java\CSW Assign
10 raised to the power 2 is: 100
```

Q8. Write a program to find the reverse of a number. For example, if the input is 123 output is 321, and if the input is-245 output is-542.



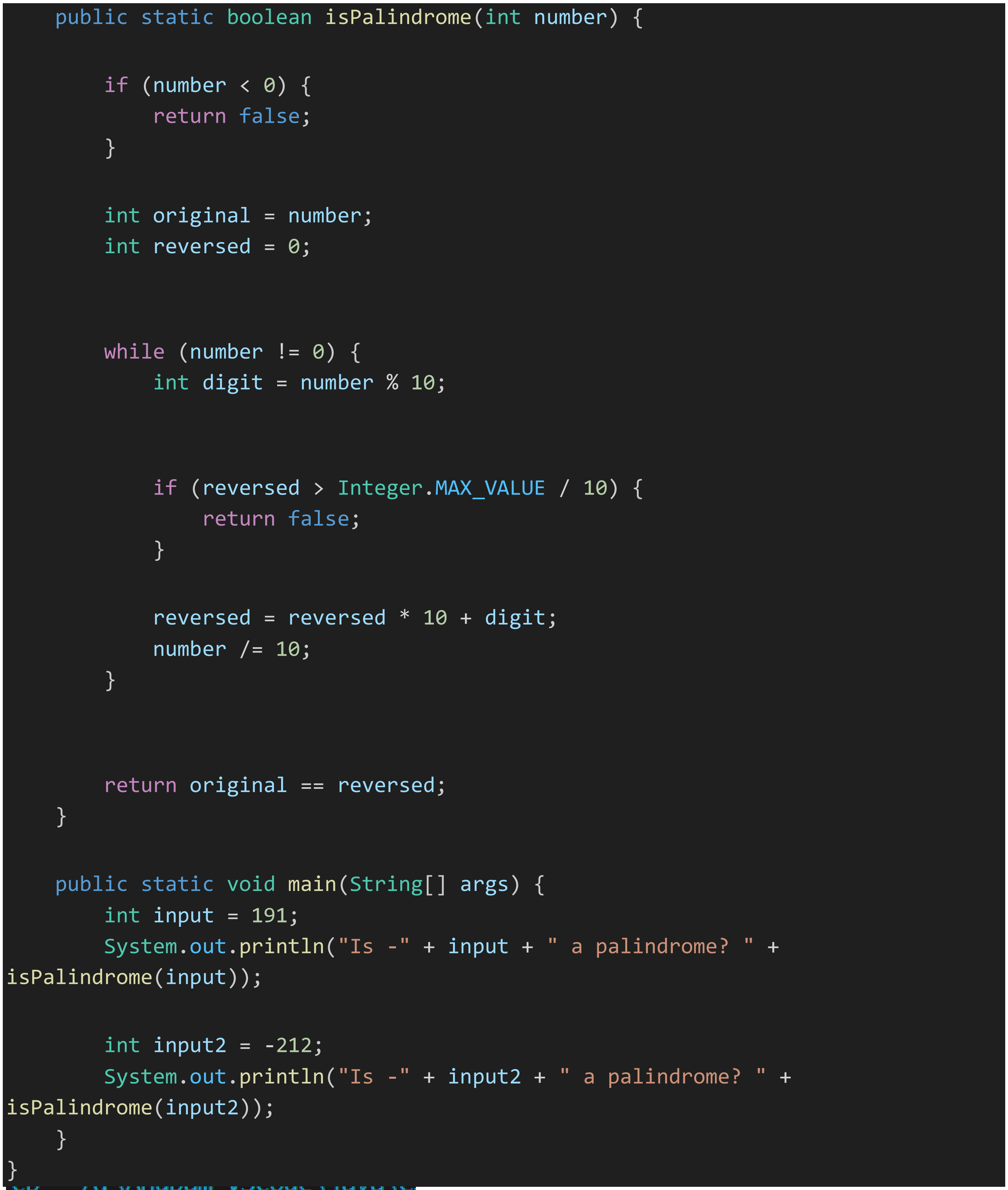
```java
public class A7_Q8 {

    public static int reverse(int number) {
        int reversed = 0;
        while (number != 0) {

            int digit = number % 10;


            if (reversed > Integer.MAX_VALUE / 10 || reversed < Integer.MIN_VALUE
/ 10) {
                throw new ArithmeticException("Overflow occurred during
reversal.");
            }


            reversed = reversed * 10 + digit;


            number /= 10;
        }

        return reversed;
    }

    public static void main(String[] args) {
        int input = -221;
        System.out.println("Original number: " + input);
        System.out.println("Reversed number: " + reverse(input));
    }
}
```



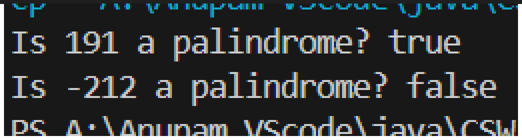
```
cp' 'A:\Anupam VScode\jav
Original number: -221
Reversed number: -122
PS A:\Anupam VScode\java\
```

Q9. Write a program to check whether a number is palindrome or not

```
public class A7_Q9 {
```

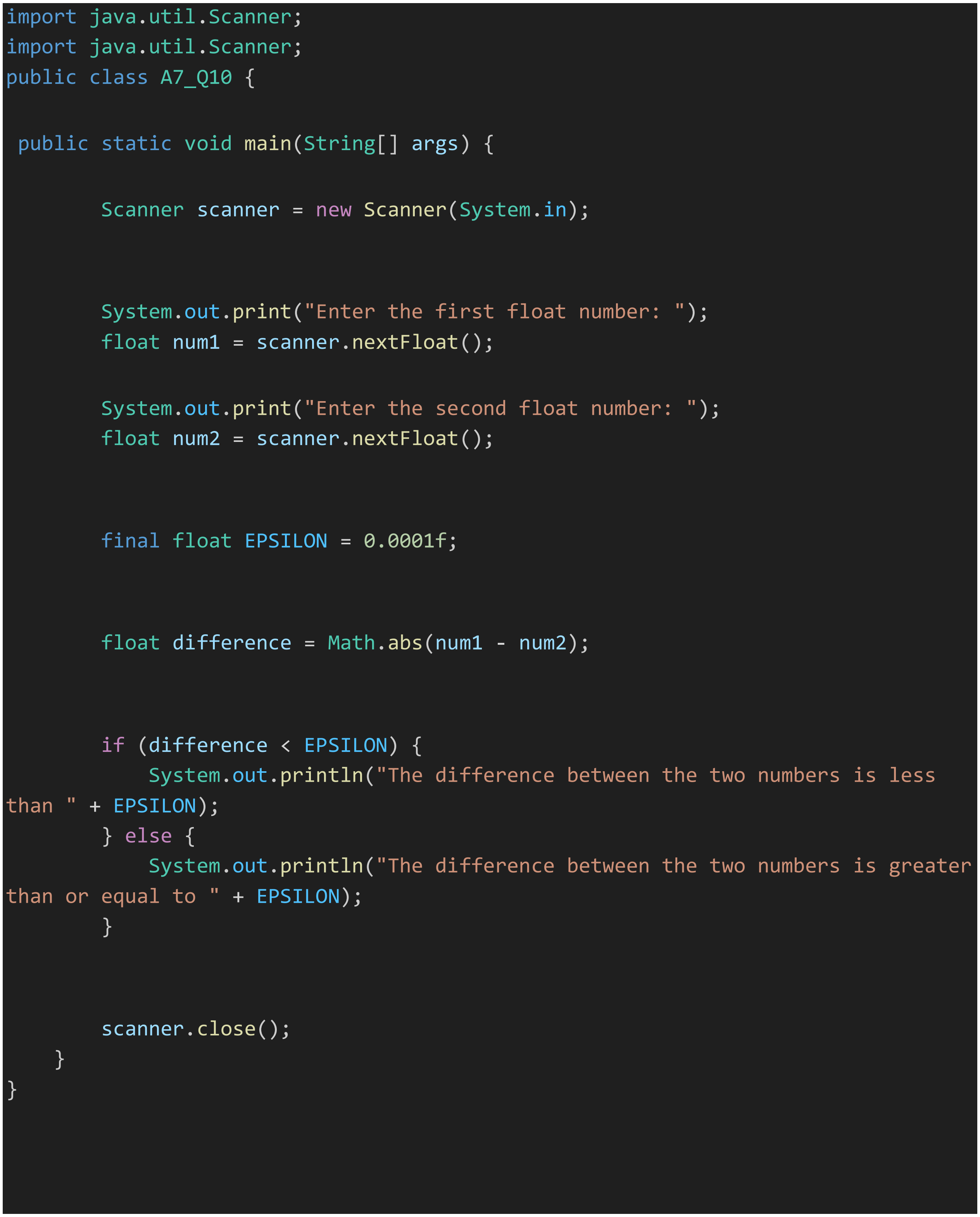


```
    public static boolean isPalindrome(int number) {

        if (number < 0) {
            return false;
        }

        int original = number;
        int reversed = 0;


        while (number != 0) {
            int digit = number % 10;


            if (reversed > Integer.MAX_VALUE / 10) {
                return false;
            }

            reversed = reversed * 10 + digit;
            number /= 10;
        }


        return original == reversed;
    }

    public static void main(String[] args) {
        int input = 191;
        System.out.println("Is -" + input + " a palindrome? " +
isPalindrome(input));

        int input2 = -212;
        System.out.println("Is -" + input2 + " a palindrome? " +
isPalindrome(input2));
    }
}
```

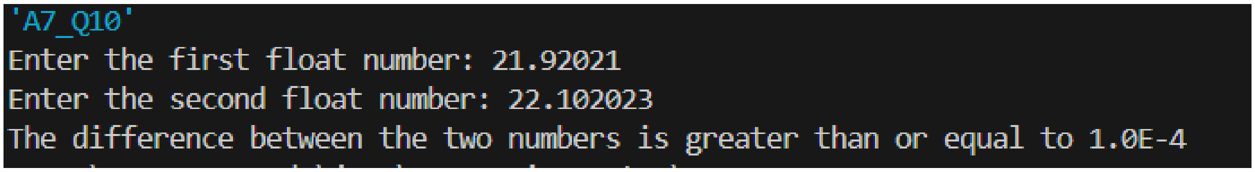


```
Is 191 a palindrome? true
Is -212 a palindrome? false
PS A:\Anupam VScode\java\CSW
```

Q10. Write a Java program that reads two float numbers and checks whether the difference between these two numbers is less than ϵ (ϵ < 1)

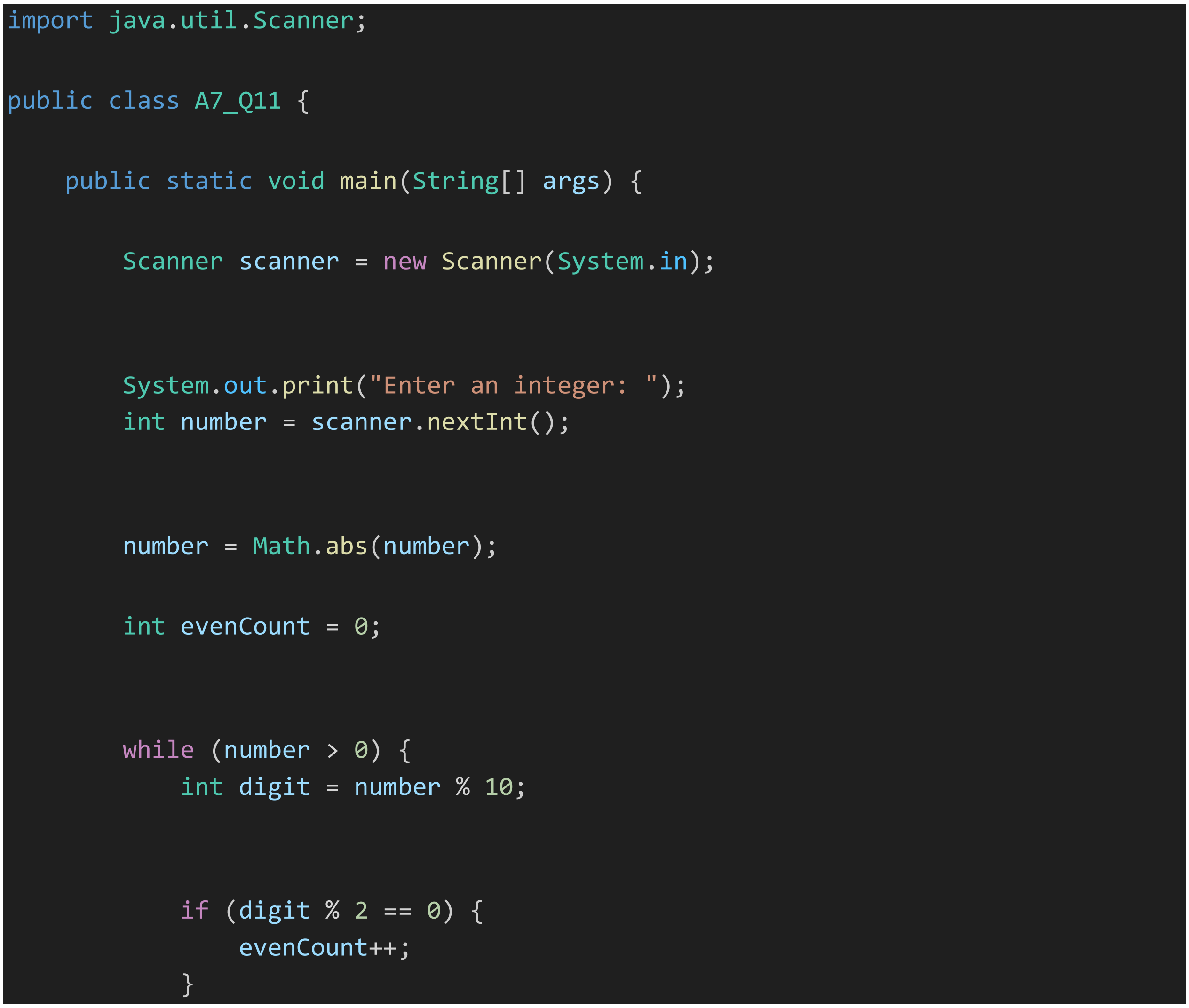


```java
import java.util.Scanner;
import java.util.Scanner;
public class A7_Q10 {

 public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter the first float number: ");
        float num1 = scanner.nextFloat();

        System.out.print("Enter the second float number: ");
        float num2 = scanner.nextFloat();


        final float EPSILON = 0.0001f;


        float difference = Math.abs(num1 - num2);


        if (difference < EPSILON) {
            System.out.println("The difference between the two numbers is less than " + EPSILON);
        } else {
            System.out.println("The difference between the two numbers is greater than or equal to " + EPSILON);
        }


        scanner.close();
    }
}
```

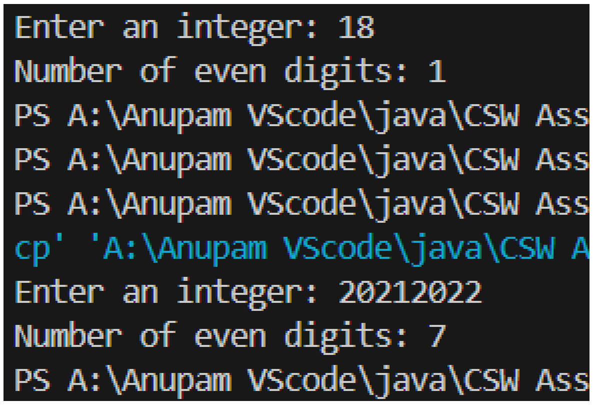


```
'A7_Q10'
Enter the first float number: 21.92021
Enter the second float number: 22.102023
The difference between the two numbers is greater than or equal to 1.0E-4
```

Q11. Write a Java program that reads an integer number and counts the number of digits that are even.



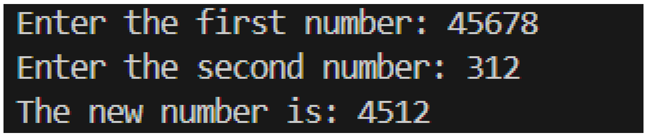
```java
import java.util.Scanner;

public class A7_Q11 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();


        number = Math.abs(number);

        int evenCount = 0;


        while (number > 0) {
            int digit = number % 10;


            if (digit % 2 == 0) {
                evenCount++;
            }
            number /= 10;
        }


        System.out.println("Number of even digits: " + evenCount);


        scanner.close();
    }
}
```



```
Enter an integer: 18
Number of even digits: 1
PS A:\Anupam VScode\java\CSW Ass
PS A:\Anupam VScode\java\CSW Ass
PS A:\Anupam VScode\java\CSW Ass
cp' 'A:\Anupam VScode\java\CSW A
Enter an integer: 20212022
Number of even digits: 7
PS A:\Anupam VScode\java\CSW Ass
```

Q12. Write a Java program that reads two integer number and create a third number by taking the first two digits of the first number and the last two digits of the second number. Example: Input: 45678, 312 Output:4512

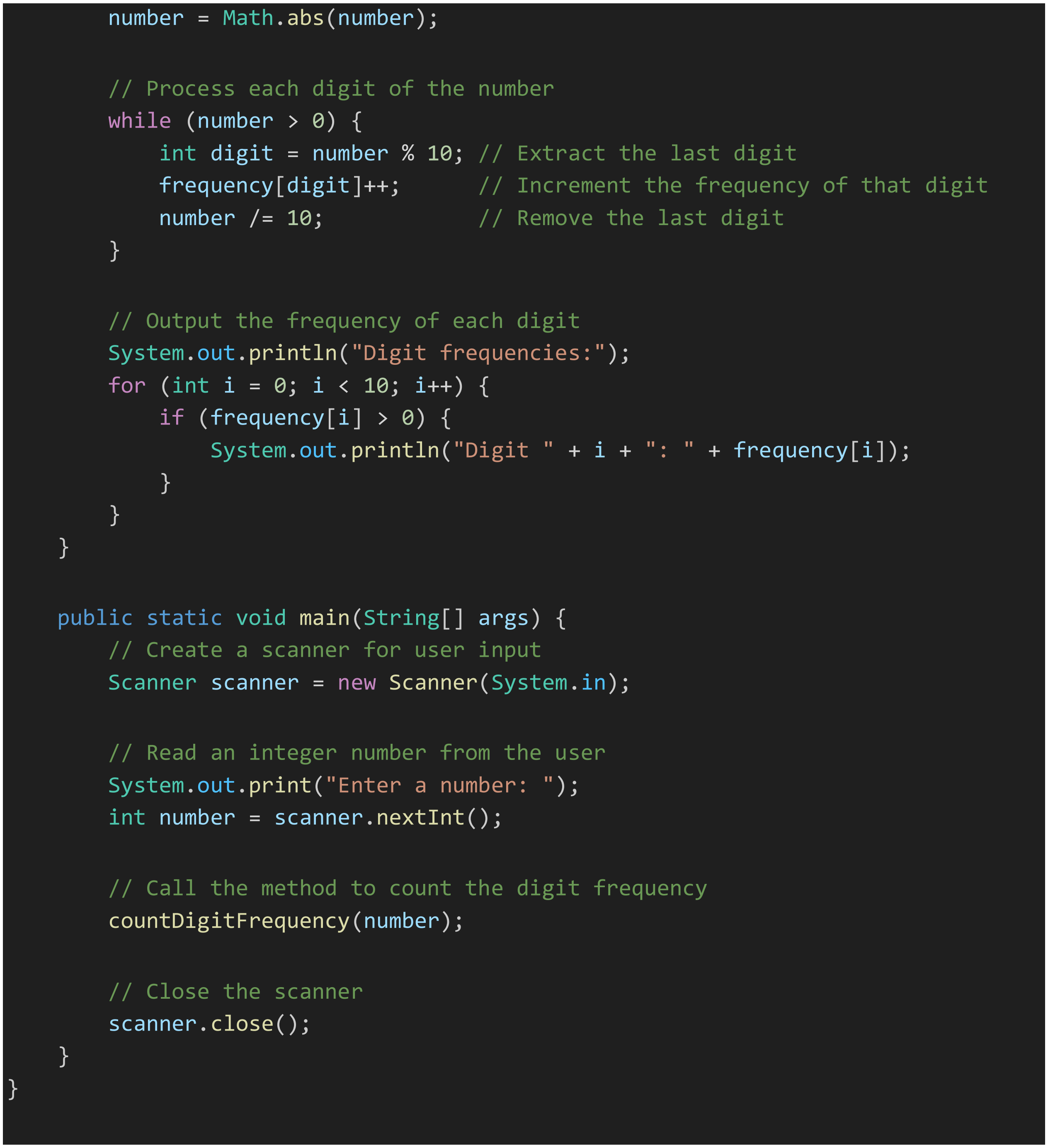
```java
import java.util.Scanner;

public class A7_Q12 {

    public static int createNumber(int num1, int num2) {

        int firstTwoDigits = num1 / (int)Math.pow(10, (int)Math.log10(num1) - 1);


        int lastTwoDigits = num2 % 100;


        return firstTwoDigits * 100 + lastTwoDigits;
    }

    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();


        int result = createNumber(num1, num2);


        System.out.println("The new number is: " + result);
        scanner.close();
    }
}
```



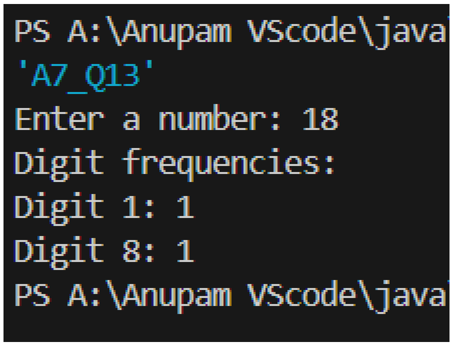
```
Enter the first number: 45678
Enter the second number: 312
The new number is: 4512
```

Q13. Write a Java program to count the frequency of each digit of a number

```java
import java.util.Scanner;

public class A7_Q13 {

    public static void countDigitFrequency(int number) {
        // Create an array to store frequency of each digit (0-9)
        int[] frequency = new int[10];

        // Handle negative numbers by converting to positive
```

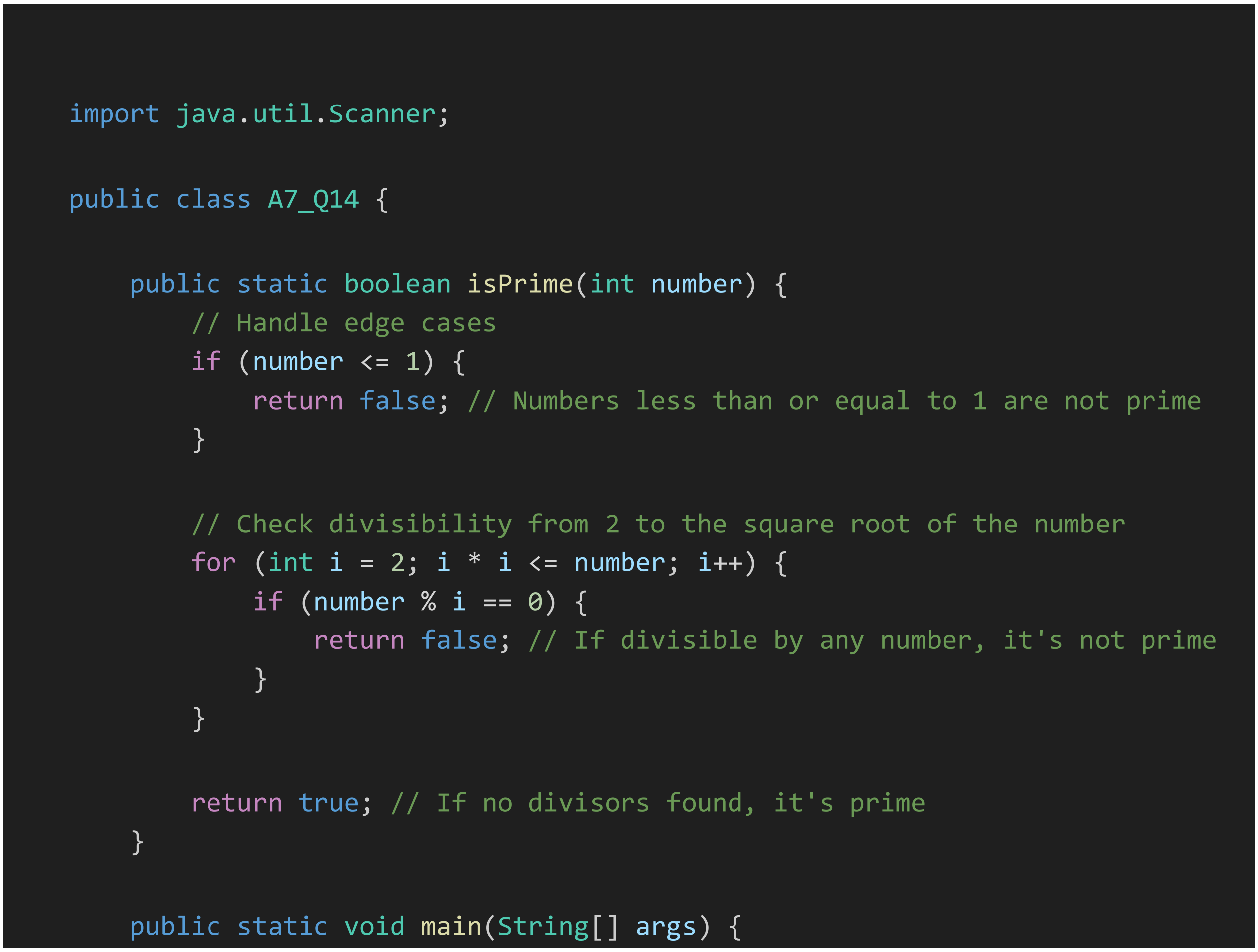


```java
        number = Math.abs(number);

        // Process each digit of the number
        while (number > 0) {
            int digit = number % 10; // Extract the last digit
            frequency[digit]++;      // Increment the frequency of that digit
            number /= 10;            // Remove the last digit
        }

        // Output the frequency of each digit
        System.out.println("Digit frequencies:");
        for (int i = 0; i < 10; i++) {
            if (frequency[i] > 0) {
                System.out.println("Digit " + i + ": " + frequency[i]);
            }
        }
    }

    public static void main(String[] args) {
        // Create a scanner for user input
        Scanner scanner = new Scanner(System.in);

        // Read an integer number from the user
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        // Call the method to count the digit frequency
        countDigitFrequency(number);

        // Close the scanner
        scanner.close();
    }
}
```

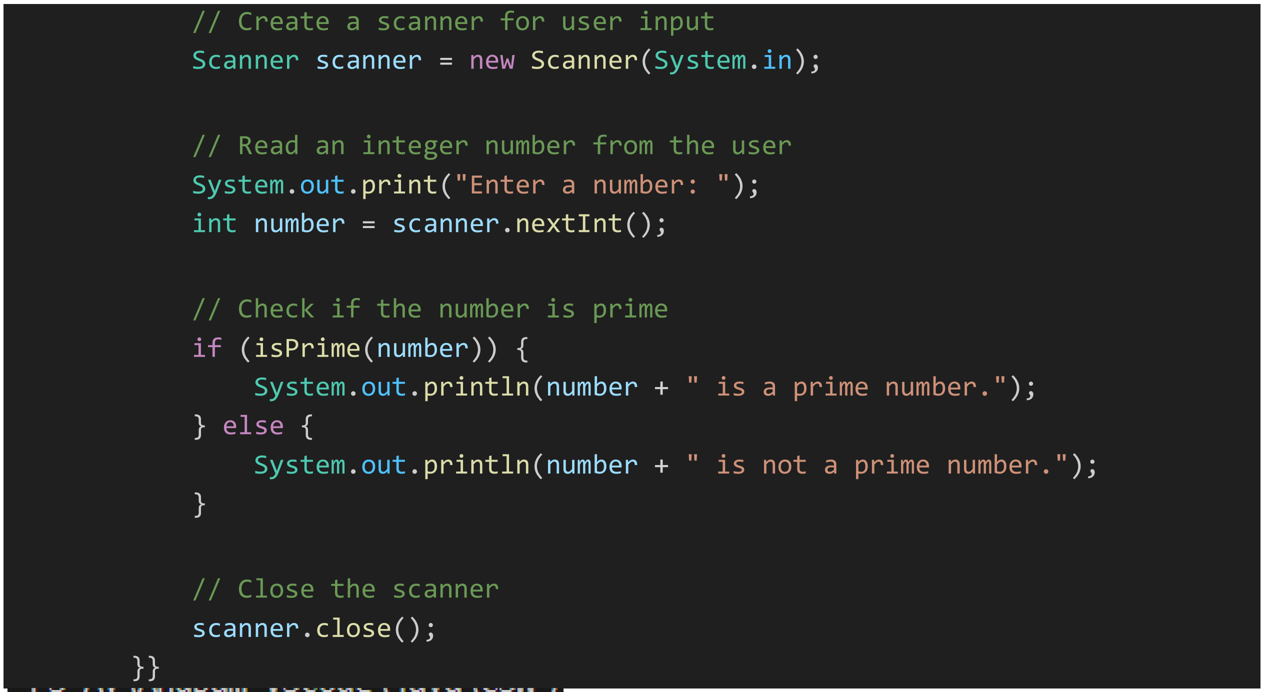


```
PS A:\Anupam VScode\java
'A7_Q13'
Enter a number: 18
Digit frequencies:
Digit 1: 1
Digit 8: 1
PS A:\Anupam VScode\java
```

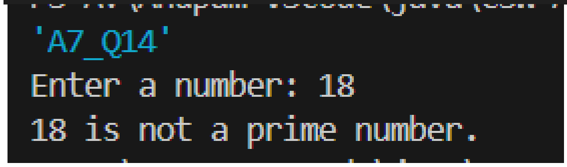
Q14. Write a Java program to check whether a number is prime or not.



```
import java.util.Scanner;

public class A7_Q14 {

    public static boolean isPrime(int number) {
        // Handle edge cases
        if (number <= 1) {
            return false; // Numbers less than or equal to 1 are not prime
        }

        // Check divisibility from 2 to the square root of the number
        for (int i = 2; i * i <= number; i++) {
            if (number % i == 0) {
                return false; // If divisible by any number, it's not prime
            }
        }

        return true; // If no divisors found, it's prime
    }

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

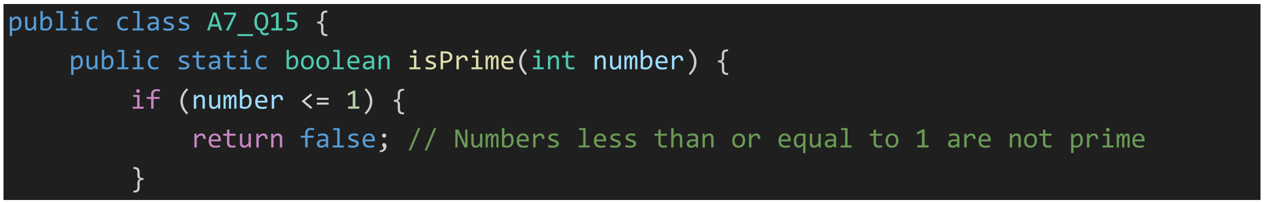


```
            // Create a scanner for user input
            Scanner scanner = new Scanner(System.in);

            // Read an integer number from the user
            System.out.print("Enter a number: ");
            int number = scanner.nextInt();

            // Check if the number is prime
            if (isPrime(number)) {
                System.out.println(number + " is a prime number.");
            } else {
                System.out.println(number + " is not a prime number.");
            }

            // Close the scanner
            scanner.close();
        }}
```

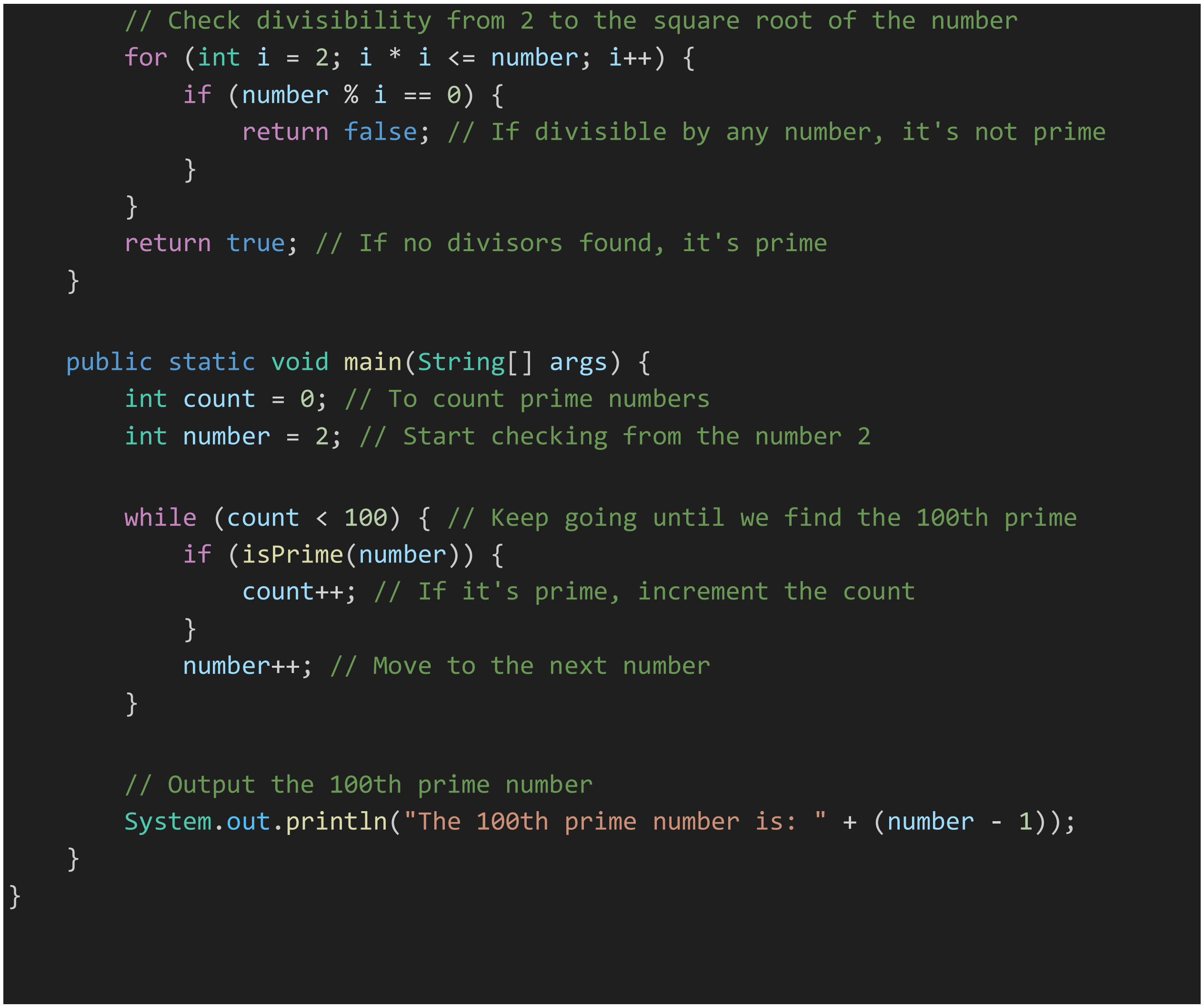


```
'A7_Q14'
Enter a number: 18
18 is not a prime number.
```

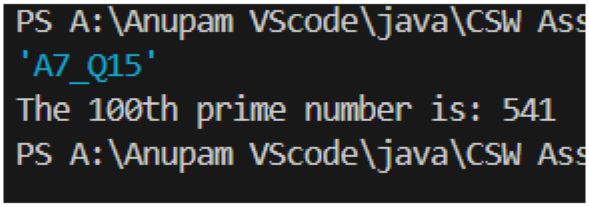
Q15. Write a program to print the first 100th prime number



```
public class A7_Q15 {
    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false; // Numbers less than or equal to 1 are not prime
        }
```

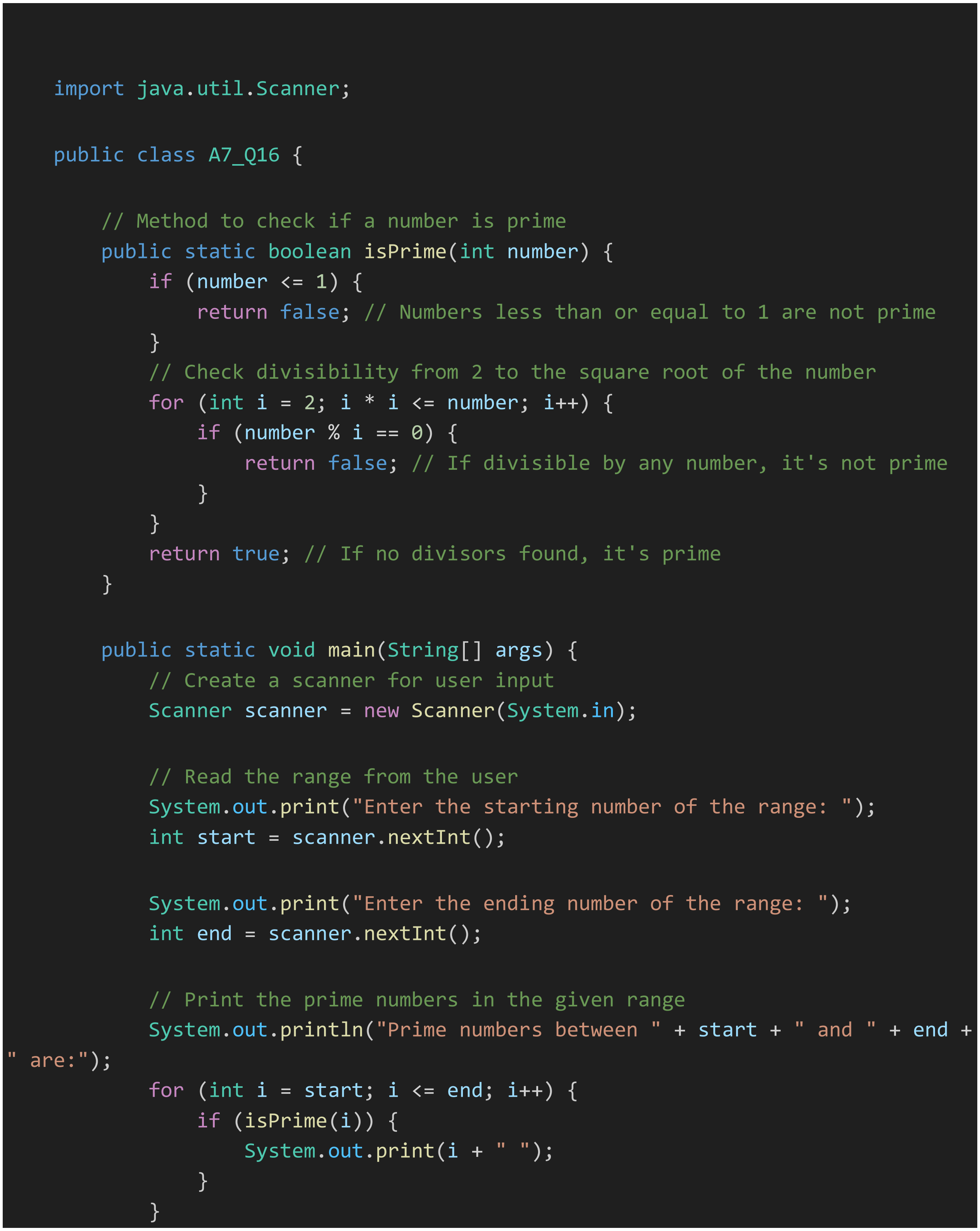


```java
        // Check divisibility from 2 to the square root of the number
        for (int i = 2; i * i <= number; i++) {
            if (number % i == 0) {
                return false; // If divisible by any number, it's not prime
            }
        }
        return true; // If no divisors found, it's prime
    }

    public static void main(String[] args) {
        int count = 0; // To count prime numbers
        int number = 2; // Start checking from the number 2

        while (count < 100) { // Keep going until we find the 100th prime
            if (isPrime(number)) {
                count++; // If it's prime, increment the count
            }
            number++; // Move to the next number
        }

        // Output the 100th prime number
        System.out.println("The 100th prime number is: " + (number - 1));
    }
}
```

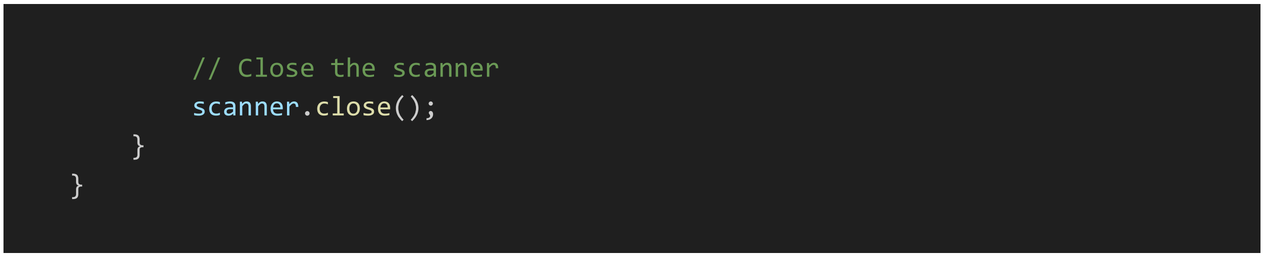


```
PS A:\Anupam VScode\java\CSW Ass
'A7_Q15'
The 100th prime number is: 541
PS A:\Anupam VScode\java\CSW Ass
```

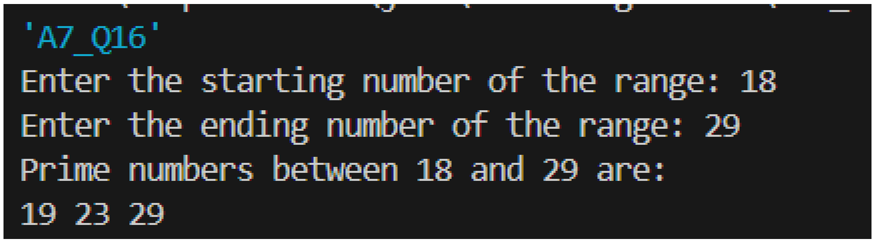
Q16. Write a Java program to print the prime number in a range



```
import java.util.Scanner;

public class A7_Q16 {

    // Method to check if a number is prime
    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false; // Numbers less than or equal to 1 are not prime
        }
        // Check divisibility from 2 to the square root of the number
        for (int i = 2; i * i <= number; i++) {
            if (number % i == 0) {
                return false; // If divisible by any number, it's not prime
            }
        }
        return true; // If no divisors found, it's prime
    }

    public static void main(String[] args) {
        // Create a scanner for user input
        Scanner scanner = new Scanner(System.in);

        // Read the range from the user
        System.out.print("Enter the starting number of the range: ");
        int start = scanner.nextInt();

        System.out.print("Enter the ending number of the range: ");
        int end = scanner.nextInt();

        // Print the prime numbers in the given range
        System.out.println("Prime numbers between " + start + " and " + end +
" are:");
        for (int i = start; i <= end; i++) {
            if (isPrime(i)) {
                System.out.print(i + " ");
            }
        }
```

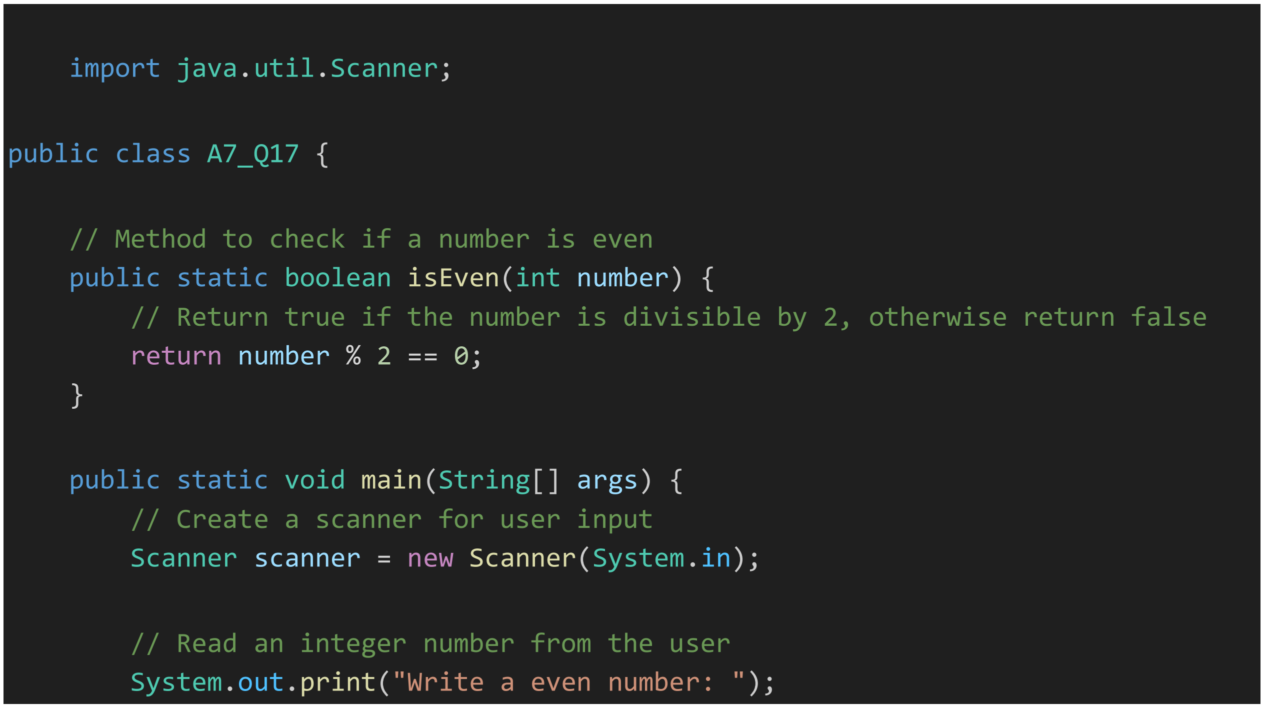


```
            // Close the scanner
            scanner.close();
        }
    }
```

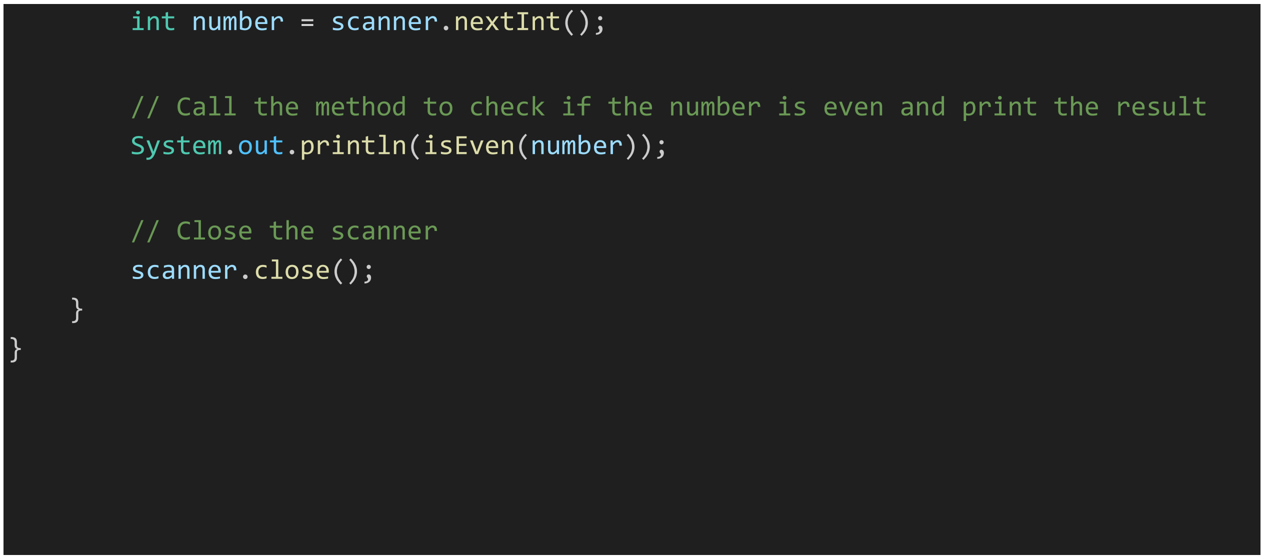


```
'A7_Q16'
Enter the starting number of the range: 18
Enter the ending number of the range: 29
Prime numbers between 18 and 29 are:
19 23 29
```

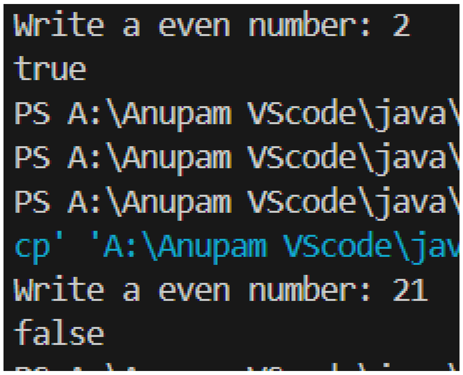
Q17. Write a program that returns true if the number is even else returns false. Note: don’t use if else.



```
    import java.util.Scanner;

public class A7_Q17 {

    // Method to check if a number is even
    public static boolean isEven(int number) {
        // Return true if the number is divisible by 2, otherwise return false
        return number % 2 == 0;
    }

    public static void main(String[] args) {
        // Create a scanner for user input
        Scanner scanner = new Scanner(System.in);

        // Read an integer number from the user
        System.out.print("Write a even number: ");
```



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

        // Call the method to check if the number is even and print the result
        System.out.println(isEven(number));

        // Close the scanner
        scanner.close();
    }
}
```



```
Write a even number: 2
true
PS A:\Anupam VScode\java\
PS A:\Anupam VScode\java\
PS A:\Anupam VScode\java\
cp' 'A:\Anupam VScode\jav
Write a even number: 21
false
```