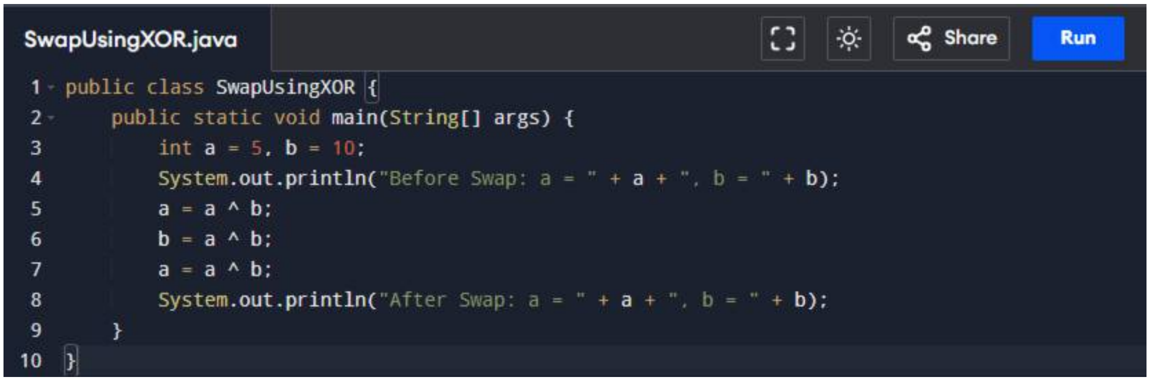
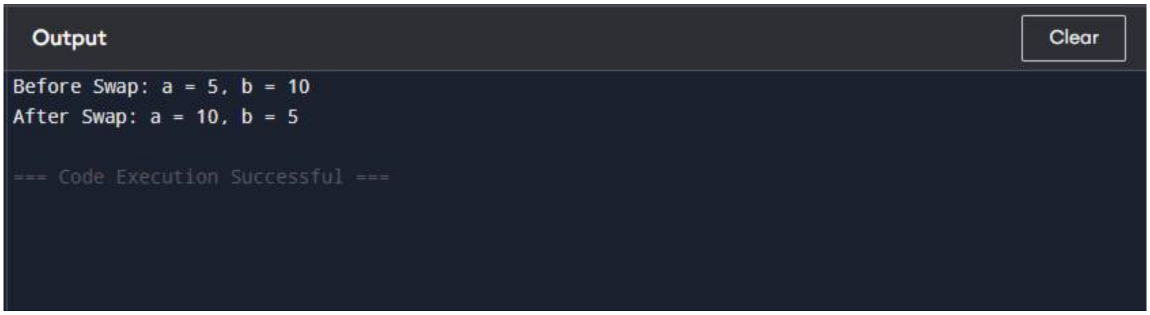
Name : Utsav Vijay Gavli

Java Asssignment 2

Q1: Write a program to swap two numbers without using a third variable and without using arithmetic operators like + or - .
Hint : Use bitwise XOR ^ operator.

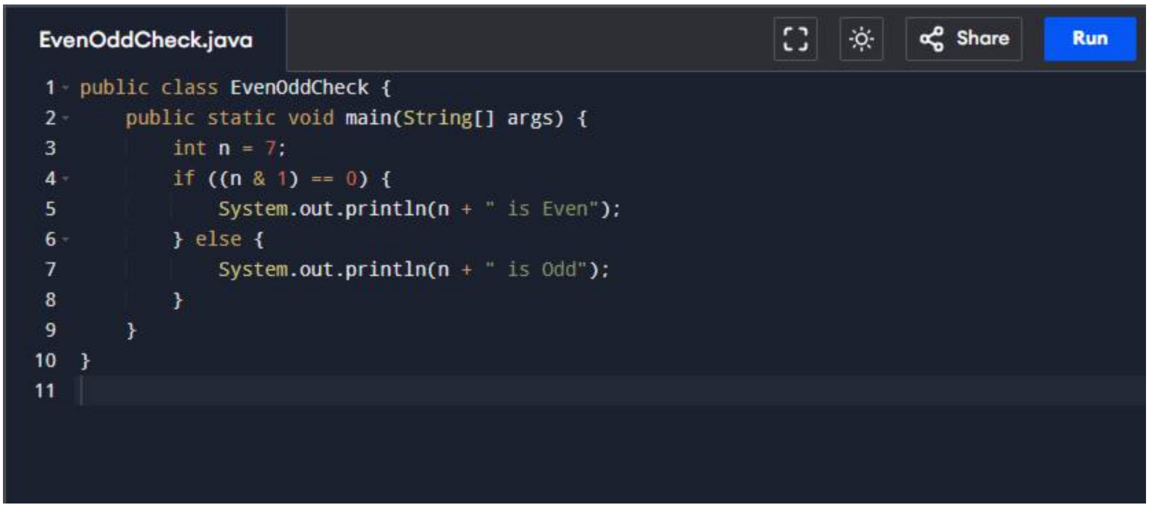


```java
public class SwapUsingXOR {
    public static void main(String[] args) {
        int a = 5, b = 10;
        System.out.println("Before Swap: a = " + a + ", b = " + b);
        a = a ^ b;
        b = a ^ b;
        a = a ^ b;
        System.out.println("After Swap: a = " + a + ", b = " + b);
    }
}
```

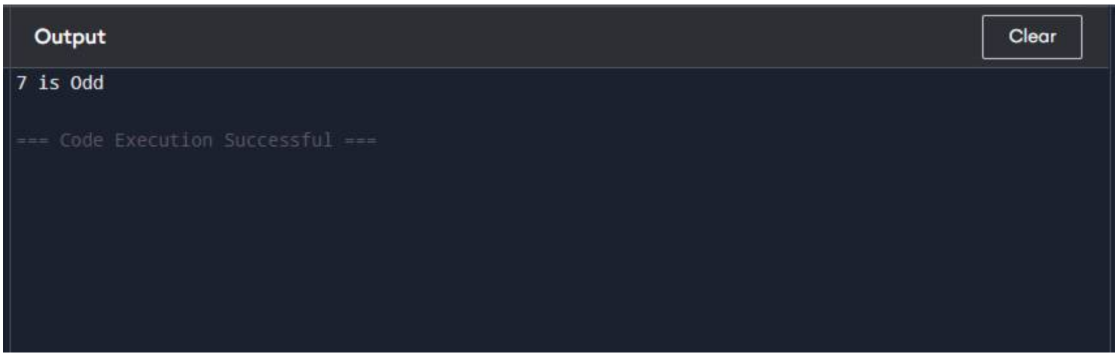


```
Before Swap: a = 5, b = 10
After Swap: a = 10, b = 5

=== Code Execution Successful ===
```

Q2: Write a program to check whether a given number is even or odd using only bitwise operators .
Hint : Use n & 1 to check.

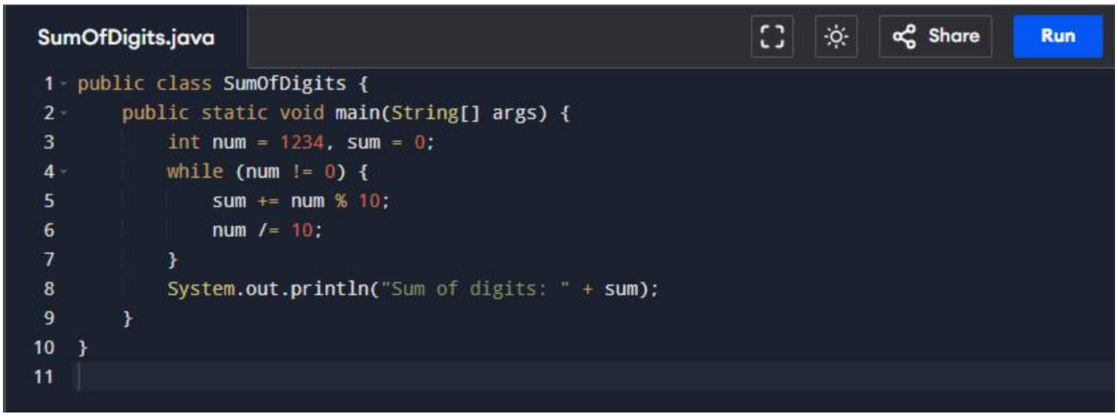


```java
public class EvenOddCheck {
    public static void main(String[] args) {
        int n = 7;
        if ((n & 1) == 0) {
            System.out.println(n + " is Even");
        } else {
            System.out.println(n + " is Odd");
        }
    }
}
```

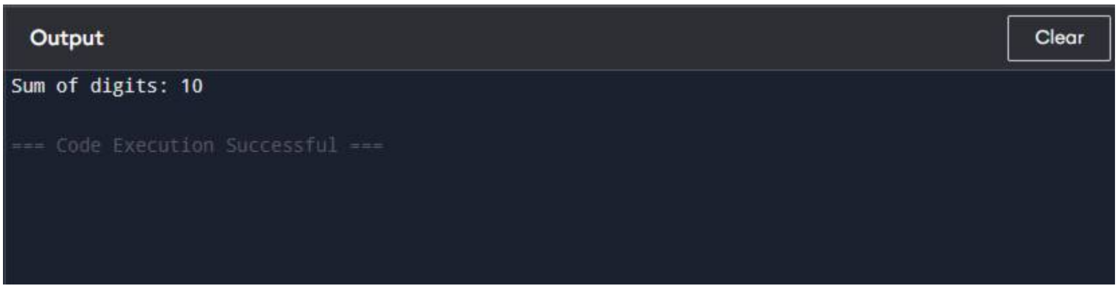


```
7 is Odd

=== Code Execution Successful ===
```

Q3: Implement a program that calculates the sum of digits of an integer using modulus (%) and division (/) operators .

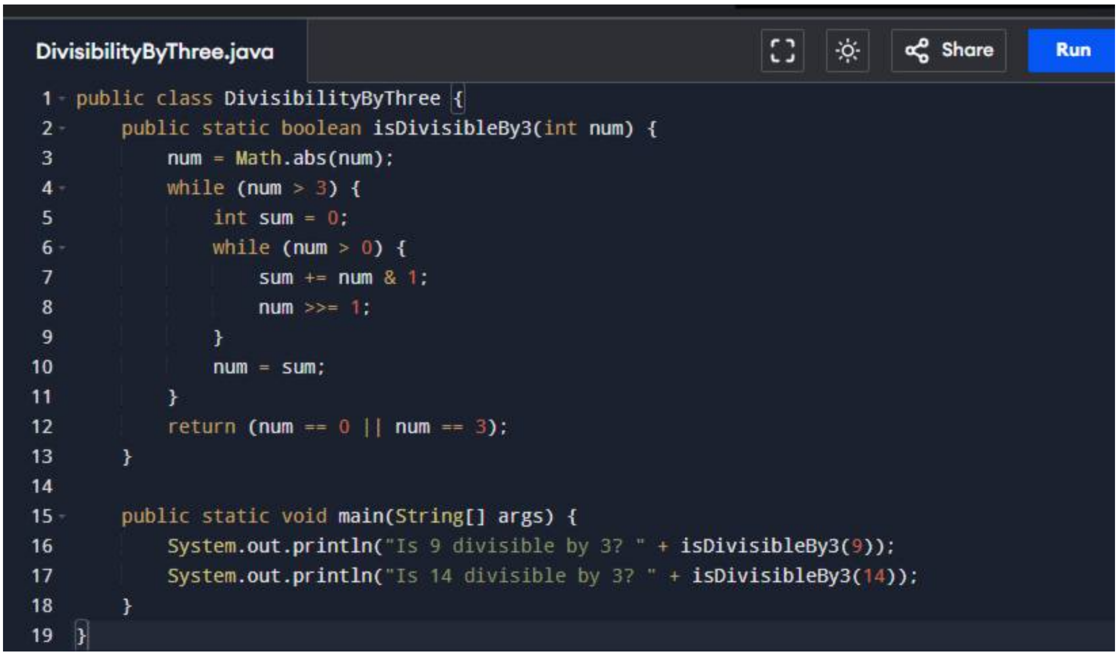


```java
public class SumOfDigits {
    public static void main(String[] args) {
        int num = 1234, sum = 0;
        while (num != 0) {
            sum += num % 10;
            num /= 10;
        }
        System.out.println("Sum of digits: " + sum);
    }
}
```

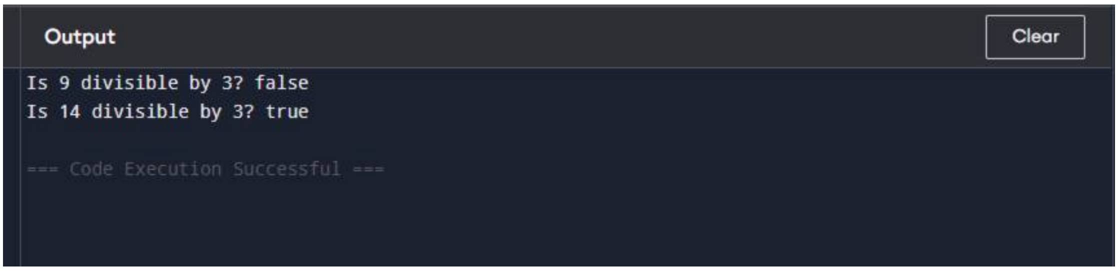


```
Sum of digits: 10

=== Code Execution Successful ===
```

Q4: Write a program to find whether a given number is divisible by 3 without using the modulus (%) or division (/) operators.
Hint : Use subtraction and bitwise shifts

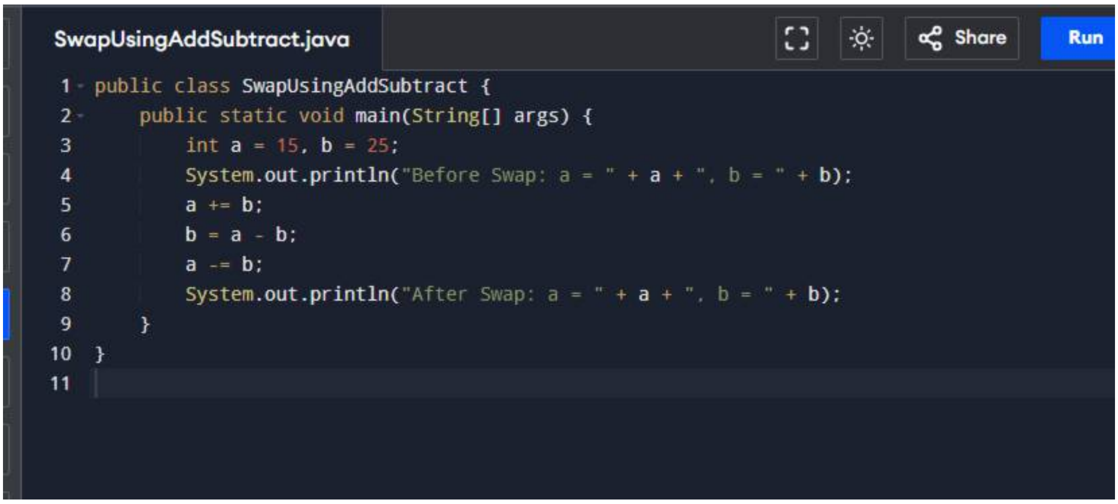


```java
public class DivisibilityByThree {
    public static boolean isDivisibleBy3(int num) {
        num = Math.abs(num);
        while (num > 3) {
            int sum = 0;
            while (num > 0) {
                sum += num & 1;
                num >>= 1;
            }
            num = sum;
        }
        return (num == 0 || num == 3);
    }

    public static void main(String[] args) {
        System.out.println("Is 9 divisible by 3? " + isDivisibleBy3(9));
        System.out.println("Is 14 divisible by 3? " + isDivisibleBy3(14));
    }
}
```

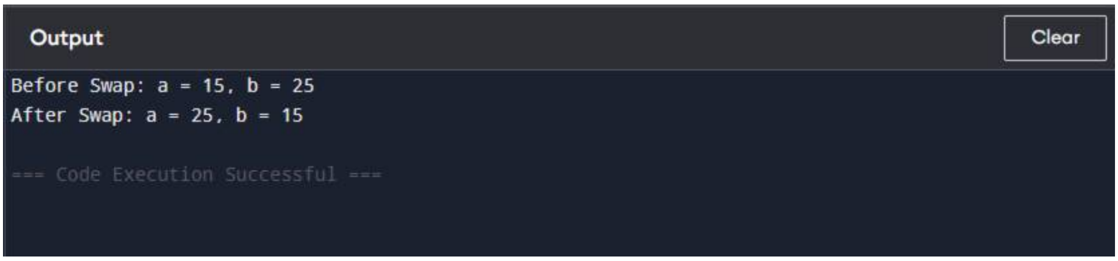


```
Is 9 divisible by 3? false
Is 14 divisible by 3? true

=== Code Execution Successful ===
```

Q5: Write a Java program to swap two numbers using the += and -= operators only.

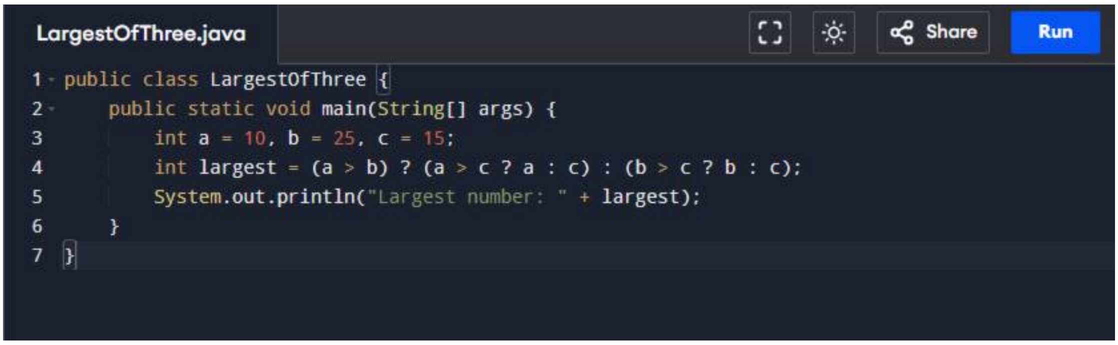


```java
public class SwapUsingAddSubtract {
    public static void main(String[] args) {
        int a = 15, b = 25;
        System.out.println("Before Swap: a = " + a + ", b = " + b);
        a += b;
        b = a - b;
        a -= b;
        System.out.println("After Swap: a = " + a + ", b = " + b);
    }
}
```

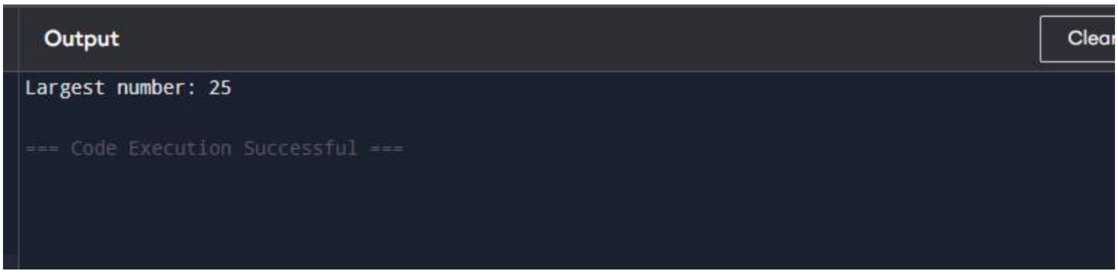


```
Before Swap: a = 15, b = 25
After Swap: a = 25, b = 15

=== Code Execution Successful ===
```

Q6: Find the largest of three numbers using the ternary operator

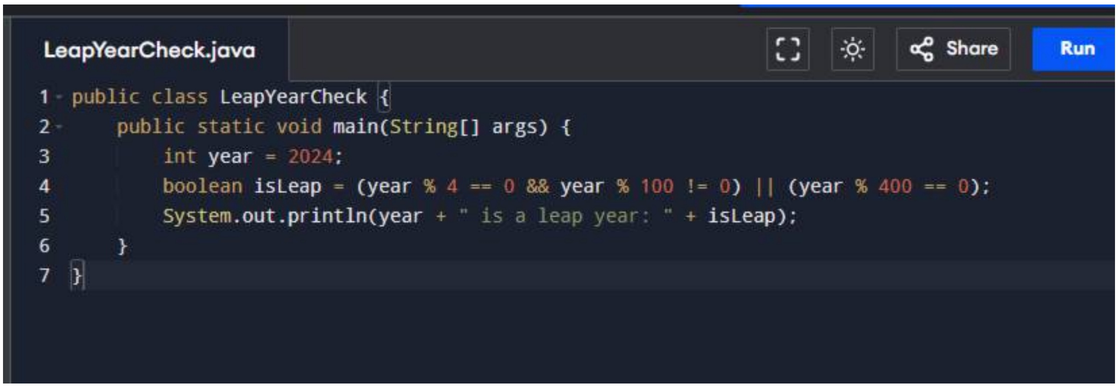


```java
public class LargestOfThree {
    public static void main(String[] args) {
        int a = 10, b = 25, c = 15;
        int largest = (a > b) ? (a > c ? a : c) : (b > c ? b : c);
        System.out.println("Largest number: " + largest);
    }
}
```

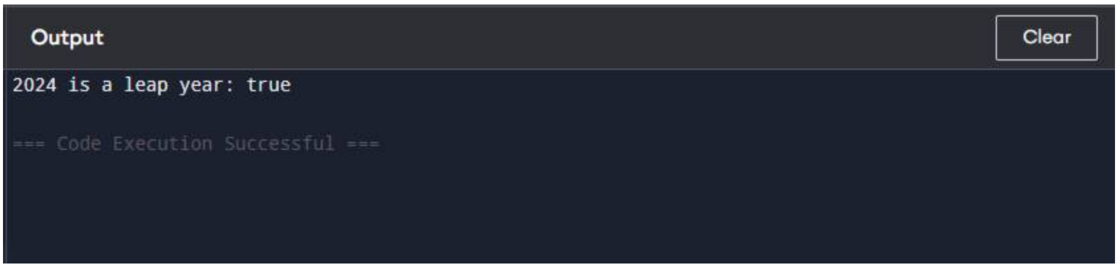


```
Largest number: 25

=== Code Execution Successful ===
```

Q7: Check if a year is a leap year using logical operators

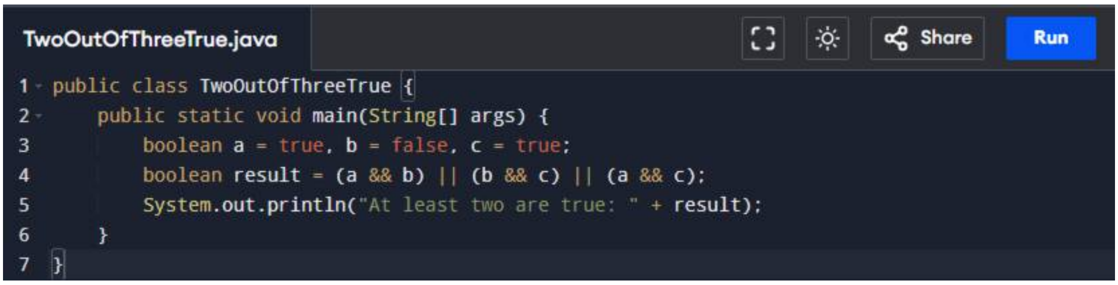


```java
public class LeapYearCheck {
    public static void main(String[] args) {
        int year = 2024;
        boolean isLeap = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
        System.out.println(year + " is a leap year: " + isLeap);
    }
}
```

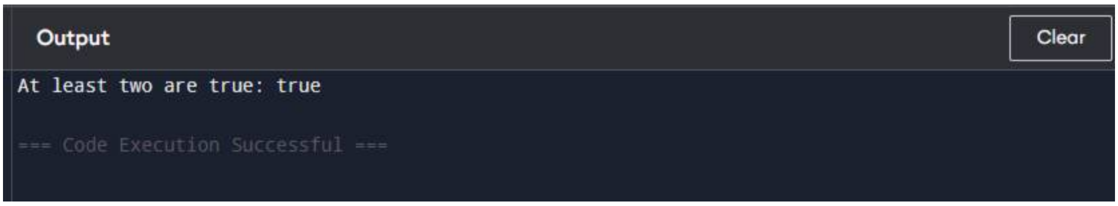


```
2024 is a leap year: true

=== Code Execution Successful ===
```

Q8: Check if at least two out of three boolean inputs are true

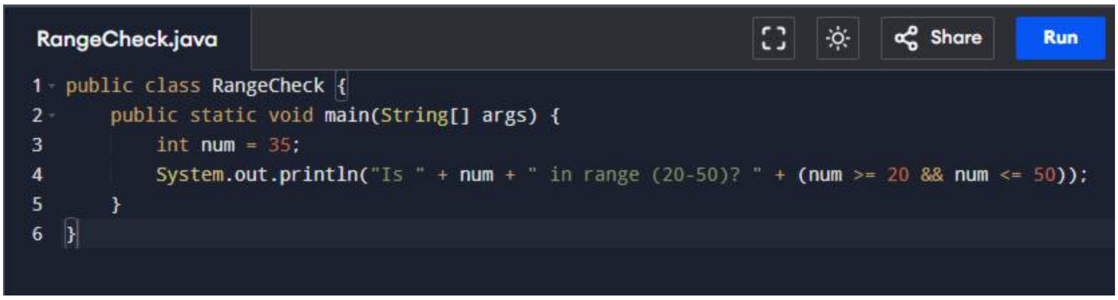


```java
public class TwoOutOfThreeTrue {
    public static void main(String[] args) {
        boolean a = true, b = false, c = true;
        boolean result = (a && b) || (b && c) || (a && c);
        System.out.println("At least two are true: " + result);
    }
}
```

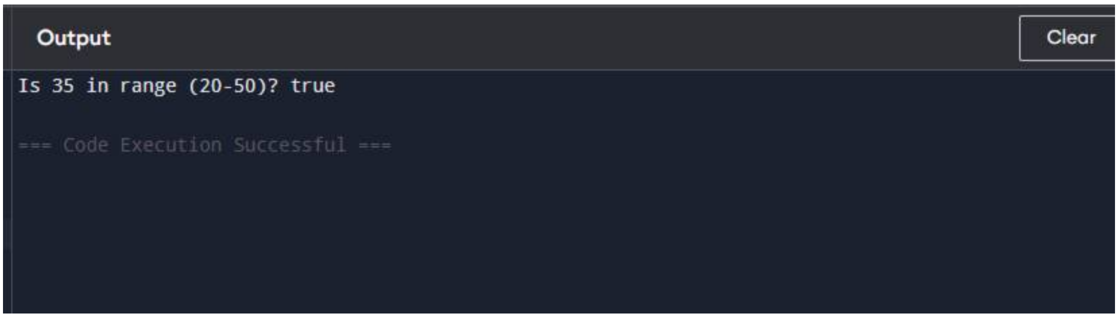


```
At least two are true: true

=== Code Execution Successful ===
```

Q9: Check if a number is within a specific range (20 to 50) without using if-else

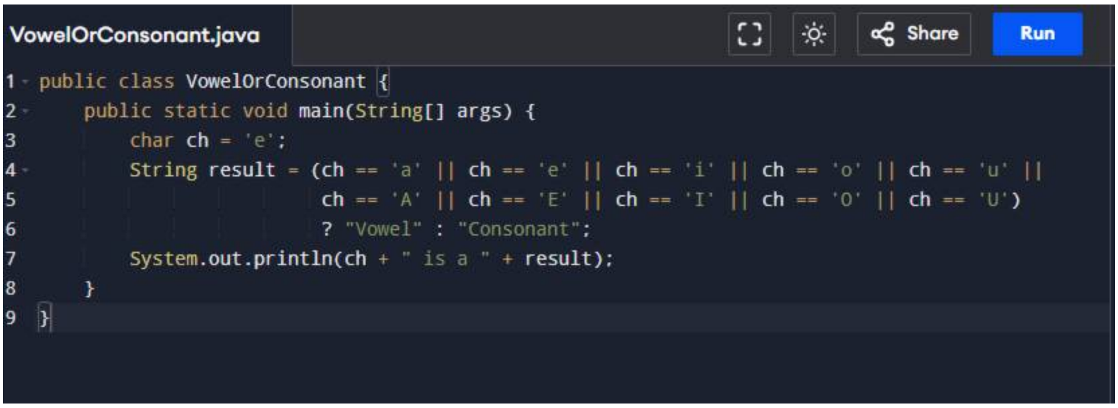


```java
public class RangeCheck {
    public static void main(String[] args) {
        int num = 35;
        System.out.println("Is " + num + " in range (20-50)? " + (num >= 20 && num <= 50));
    }
}
```

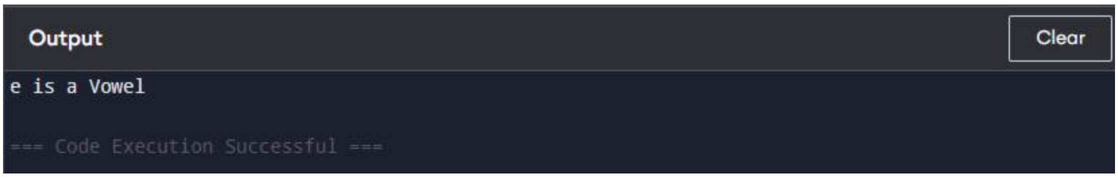


```
Is 35 in range (20-50)? true

=== Code Execution Successful ===
```

Q10: Check if a character is a vowel or consonant using the ternary operator

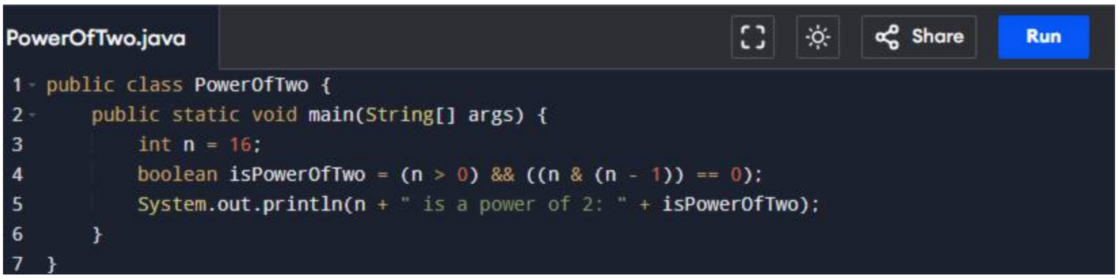


```java
public class VowelOrConsonant {
    public static void main(String[] args) {
        char ch = 'e';
        String result = (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
                         ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
                         ? "Vowel" : "Consonant";
        System.out.println(ch + " is a " + result);
    }
}
```

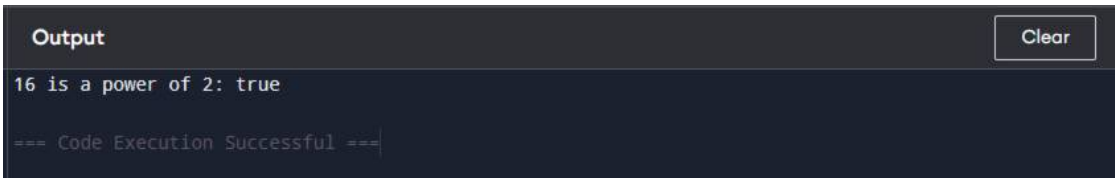


```
e is a Vowel

=== Code Execution Successful ===
```

Q11: Check if a number is a power of 2 using bitwise operators

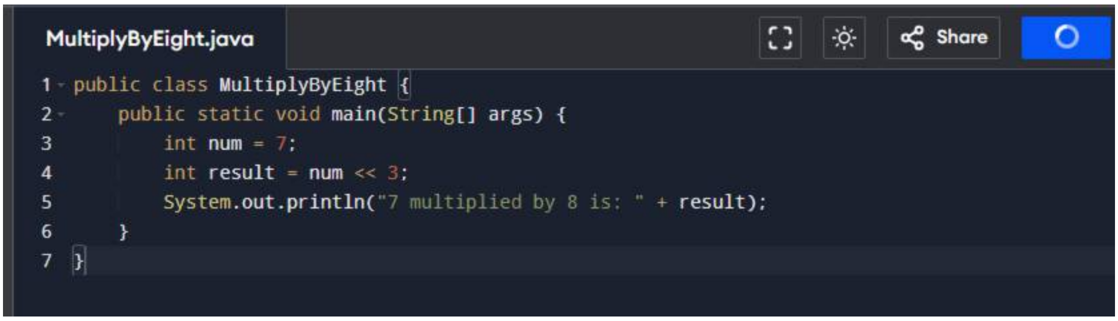


```java
public class PowerOfTwo {
    public static void main(String[] args) {
        int n = 16;
        boolean isPowerOfTwo = (n > 0) && ((n & (n - 1)) == 0);
        System.out.println(n + " is a power of 2: " + isPowerOfTwo);
    }
}
```

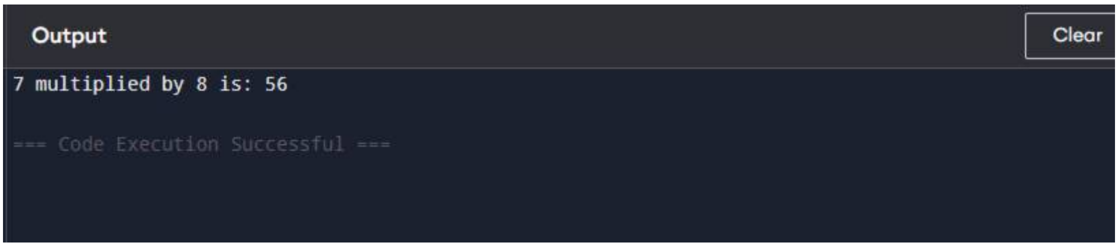


```
16 is a power of 2: true

=== Code Execution Successful ===
```

Q12: Multiply a number by 8 without using * or /

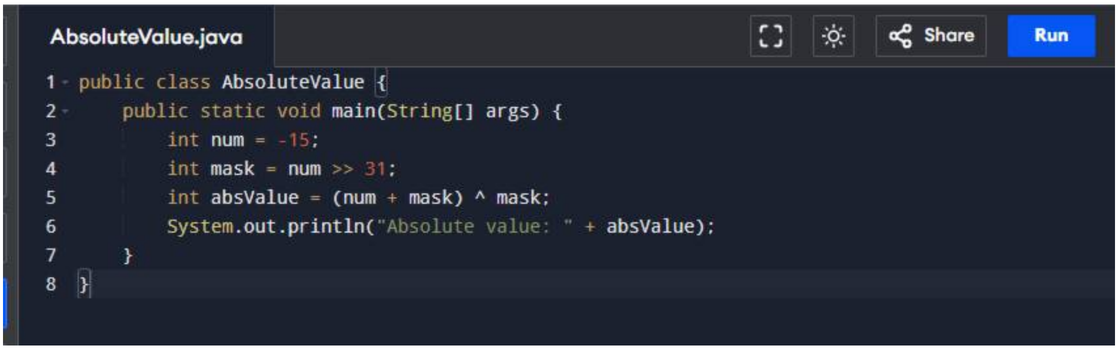


```java
public class MultiplyByEight {
    public static void main(String[] args) {
        int num = 7;
        int result = num << 3;
        System.out.println("7 multiplied by 8 is: " + result);
    }
}
```

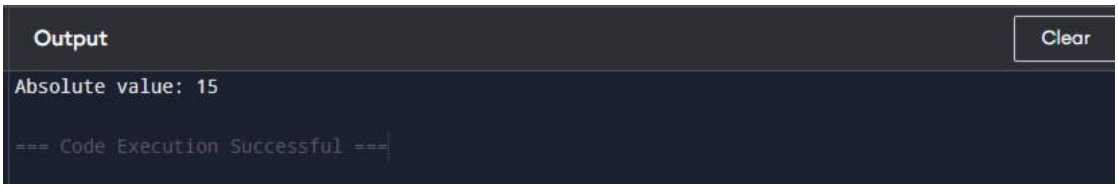


```
7 multiplied by 8 is: 56

=== Code Execution Successful ===
```

Q13: Find the absolute value of an integer using bitwise operators

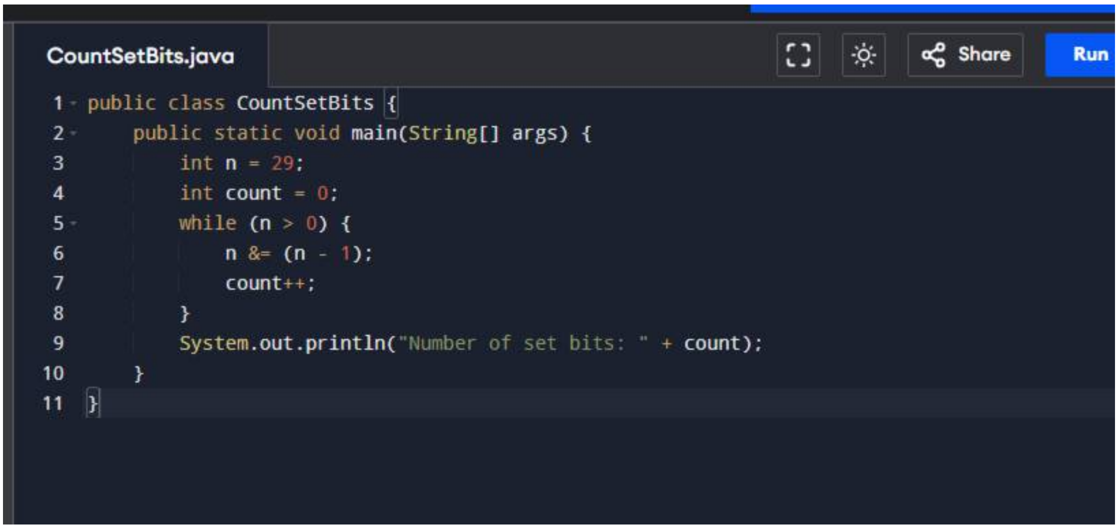


```java
public class AbsoluteValue {
    public static void main(String[] args) {
        int num = -15;
        int mask = num >> 31;
        int absValue = (num + mask) ^ mask;
        System.out.println("Absolute value: " + absValue);
    }
}
```

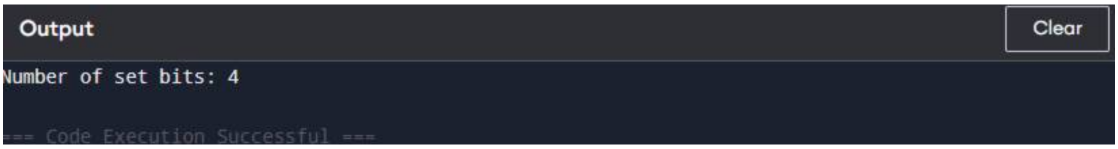


```
Absolute value: 15

=== Code Execution Successful ===
```

Q14: Count the number of 1s in a binary representation of a number

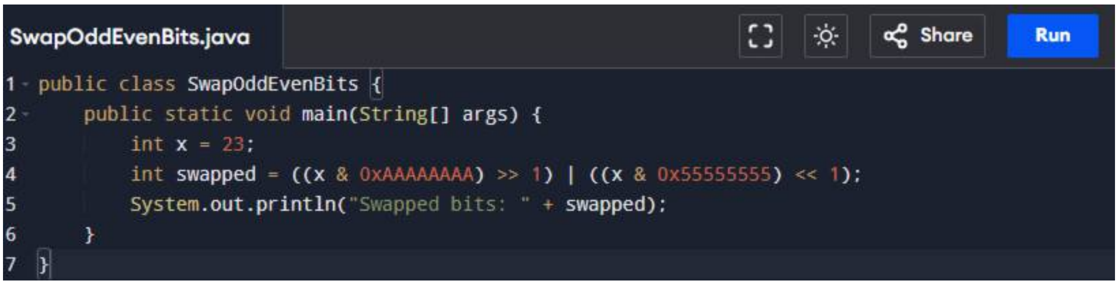


```java
public class CountSetBits {
    public static void main(String[] args) {
        int n = 29;
        int count = 0;
        while (n > 0) {
            n &= (n - 1);
            count++;
        }
        System.out.println("Number of set bits: " + count);
    }
}
```

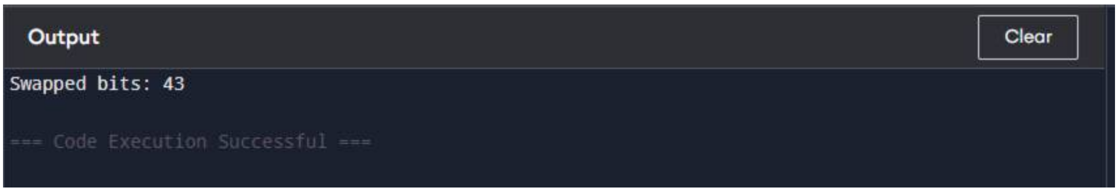


```
Number of set bits: 4

=== Code Execution Successful ===
```

Q15: Swap odd and even bits of a number using bitwise operators

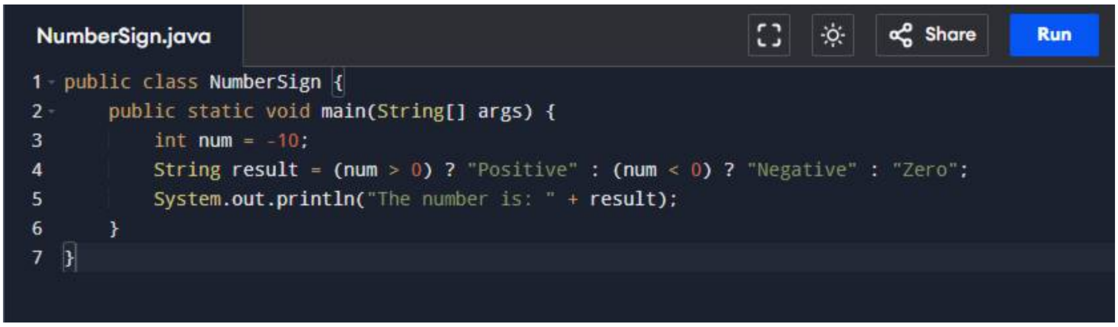


```java
public class SwapOddEvenBits {
    public static void main(String[] args) {
        int x = 23;
        int swapped = ((x & 0xAAAAAAAA) >> 1) | ((x & 0x55555555) << 1);
        System.out.println("Swapped bits: " + swapped);
    }
}
```

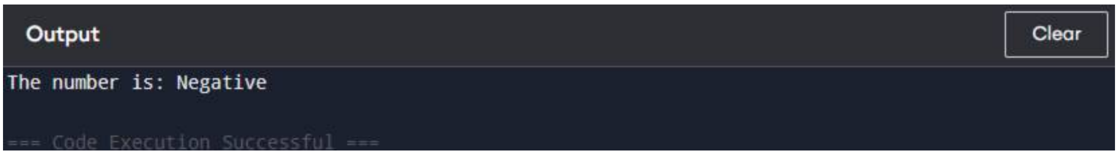


```
Swapped bits: 43

=== Code Execution Successful ===
```

Q16: Determine if a number is positive, negative, or zero using the ternary operator

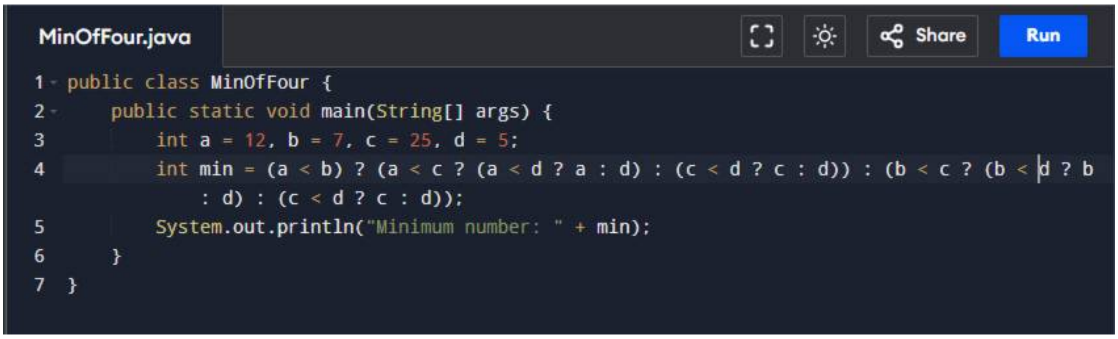


```java
public class NumberSign {
    public static void main(String[] args) {
        int num = -10;
        String result = (num > 0) ? "Positive" : (num < 0) ? "Negative" : "Zero";
        System.out.println("The number is: " + result);
    }
}
```

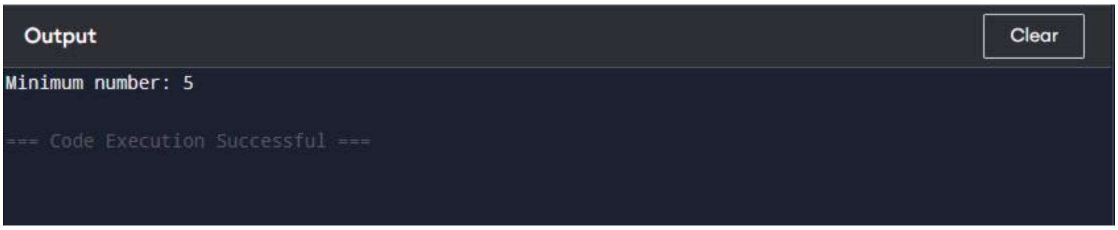


```
The number is: Negative

=== Code Execution Successful ===
```

Q17: Find the minimum of four numbers using nested ternary operators

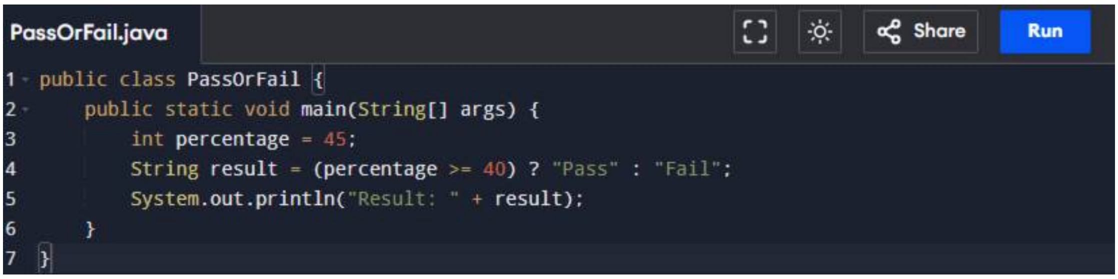


```java
public class MinOfFour {
    public static void main(String[] args) {
        int a = 12, b = 7, c = 25, d = 5;
        int min = (a < b) ? (a < c ? (a < d ? a : d) : (c < d ? c : d)) : (b < c ? (b < d ? b : d) : (c < d ? c : d));
        System.out.println("Minimum number: " + min);
    }
}
```

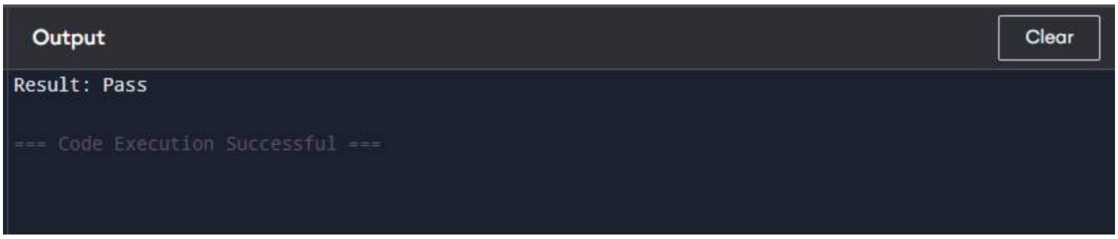


```
Minimum number: 5

=== Code Execution Successful ===
```

Q18: Print "Pass" if percentage is 40 or above; otherwise, print "Fail" using the ternary operator

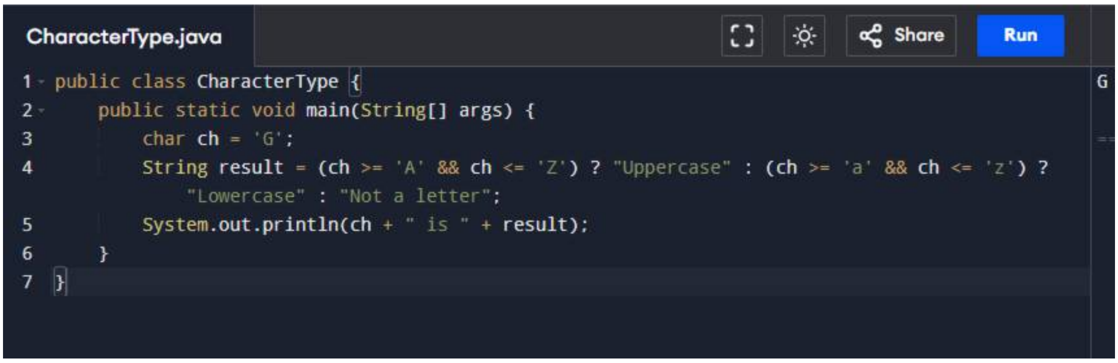


```java
public class PassOrFail {
    public static void main(String[] args) {
        int percentage = 45;
        String result = (percentage >= 40) ? "Pass" : "Fail";
        System.out.println("Result: " + result);
    }
}
```

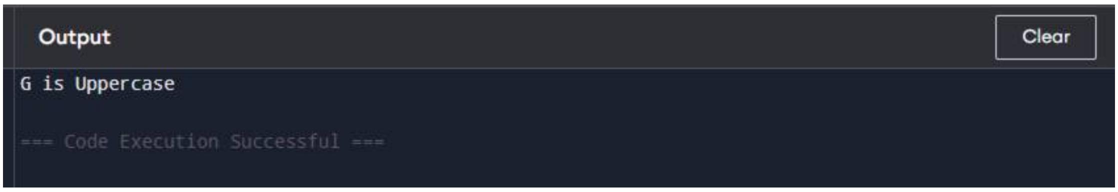


```
Result: Pass

=== Code Execution Successful ===
```

Q19: Check if a character is uppercase, lowercase, or not a letter using the ternary operator

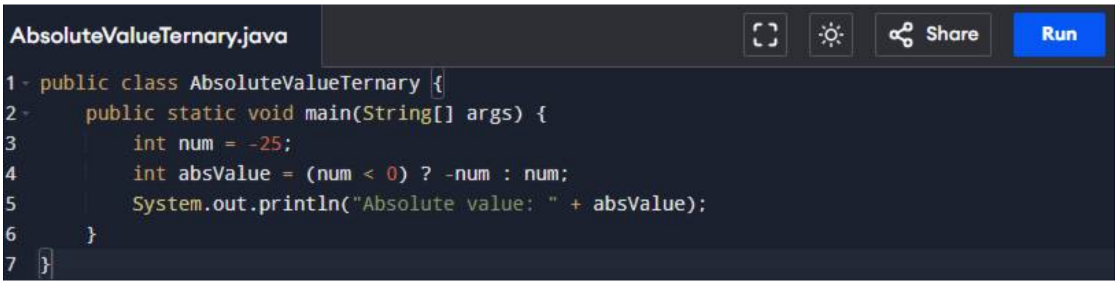


```java
public class CharacterType {
    public static void main(String[] args) {
        char ch = 'G';
        String result = (ch >= 'A' && ch <= 'Z') ? "Uppercase" : (ch >= 'a' && ch <= 'z') ? "Lowercase" : "Not a letter";
        System.out.println(ch + " is " + result);
    }
}
```

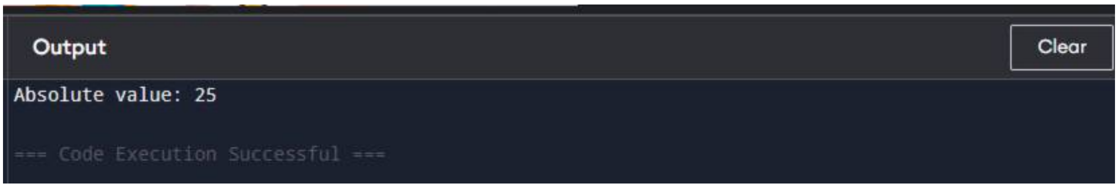


```
G is Uppercase

=== Code Execution Successful ===
```

Q20: Find the absolute value of a number using the ternary operator

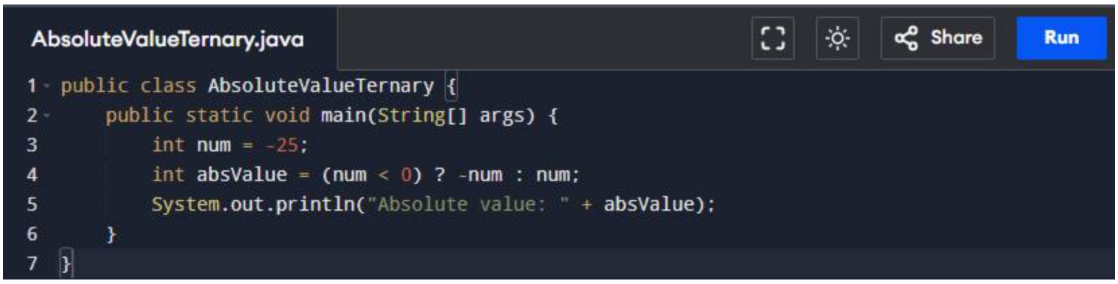


```java
public class AbsoluteValueTernary {
    public static void main(String[] args) {
        int num = -25;
        int absValue = (num < 0) ? -num : num;
        System.out.println("Absolute value: " + absValue);
    }
}
```

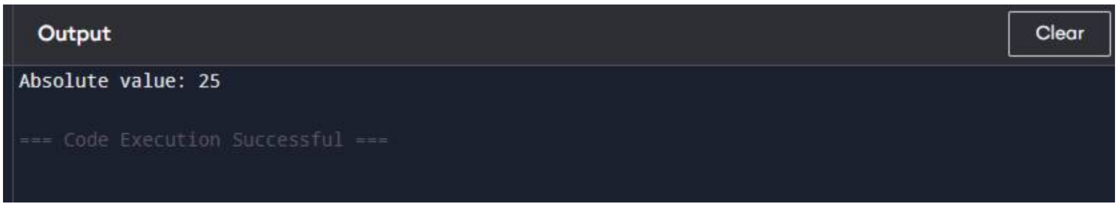


```
Absolute value: 25

=== Code Execution Successful ===
```

// Q21: Increment a number without using + or ++ operators

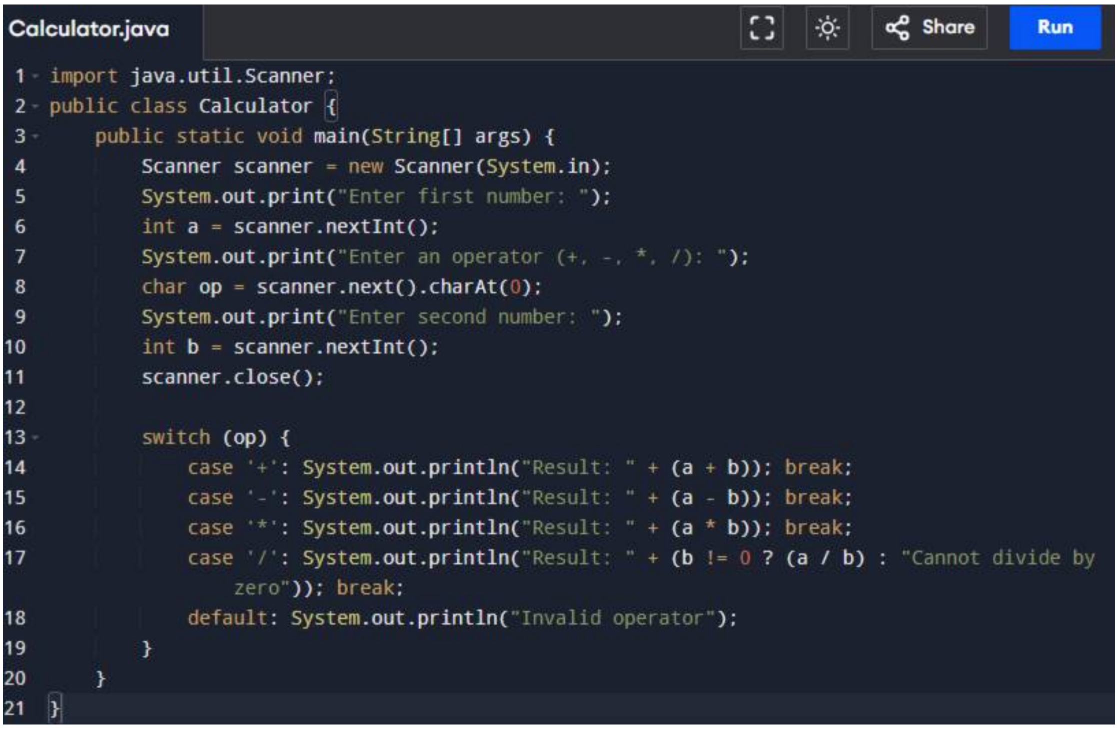


```java
public class AbsoluteValueTernary {
    public static void main(String[] args) {
        int num = -25;
        int absValue = (num < 0) ? -num : num;
        System.out.println("Absolute value: " + absValue);
    }
}
```

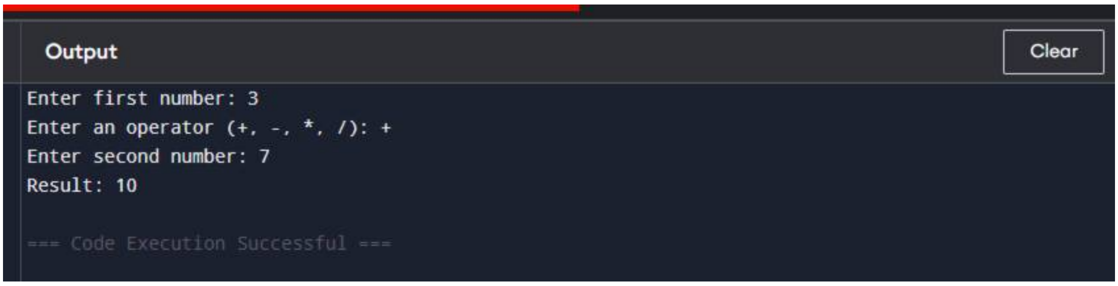


```
Absolute value: 25

=== Code Execution Successful ===
```

Q22: Simple calculator using switch-case

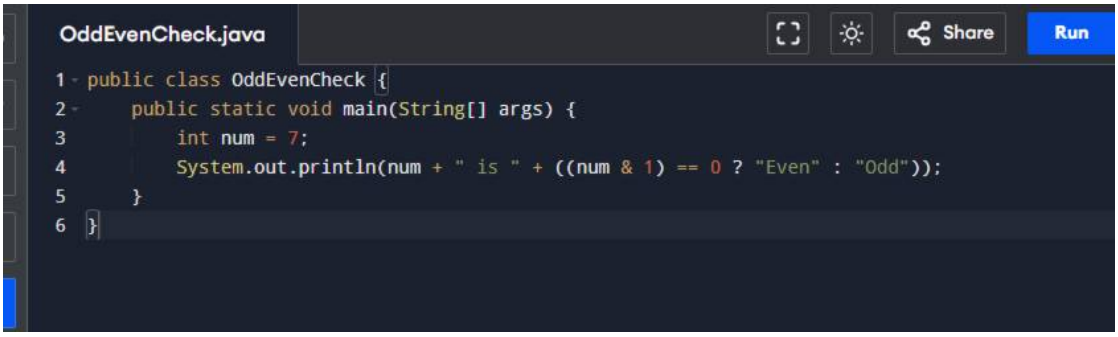


```java
import java.util.Scanner;
public class Calculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first number: ");
        int a = scanner.nextInt();
        System.out.print("Enter an operator (+, -, *, /): ");
        char op = scanner.next().charAt(0);
        System.out.print("Enter second number: ");
        int b = scanner.nextInt();
        scanner.close();

        switch (op) {
            case '+': System.out.println("Result: " + (a + b)); break;
            case '-': System.out.println("Result: " + (a - b)); break;
            case '*': System.out.println("Result: " + (a * b)); break;
            case '/': System.out.println("Result: " + (b != 0 ? (a / b) : "Cannot divide by zero")); break;
            default: System.out.println("Invalid operator");
        }
    }
}
```

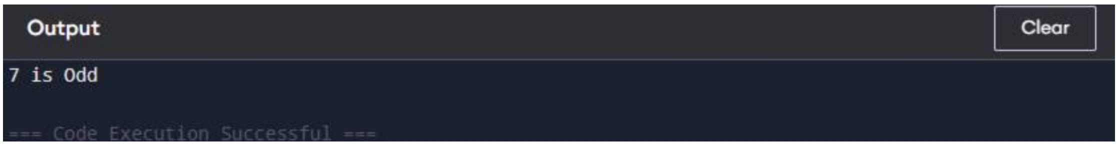


```
Enter first number: 3
Enter an operator (+, -, *, /): +
Enter second number: 7
Result: 10

=== Code Execution Successful ===
```

Q23: Check if a number is odd or even using bitwise & operator

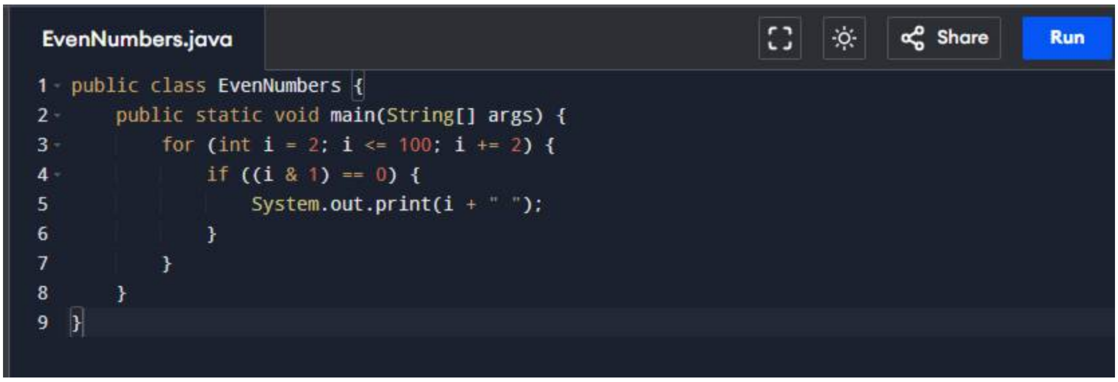


```java
public class OddEvenCheck {
    public static void main(String[] args) {
        int num = 7;
        System.out.println(num + " is " + ((num & 1) == 0 ? "Even" : "Odd"));
    }
}
```

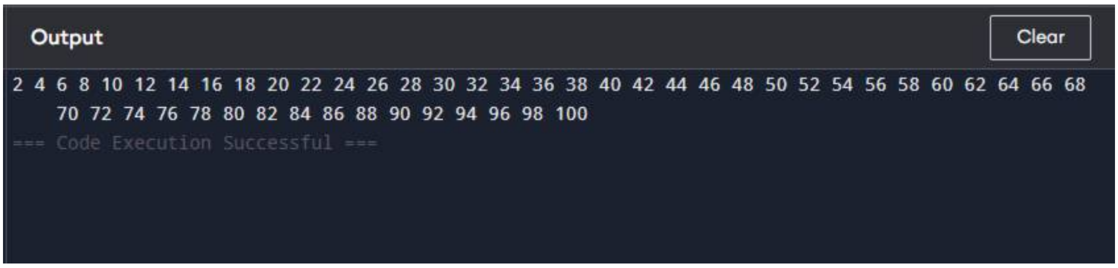


```
7 is Odd

=== Code Execution Successful ===
```

Q24: Print all even numbers from 1 to 100 using bitwise AND and for loop

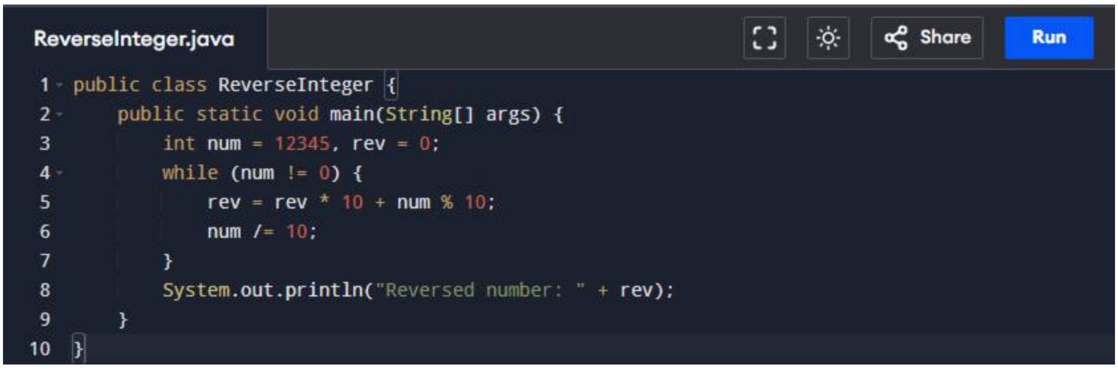


```java
public class EvenNumbers {
    public static void main(String[] args) {
        for (int i = 2; i <= 100; i += 2) {
            if ((i & 1) == 0) {
                System.out.print(i + " ");
            }
        }
    }
}
```

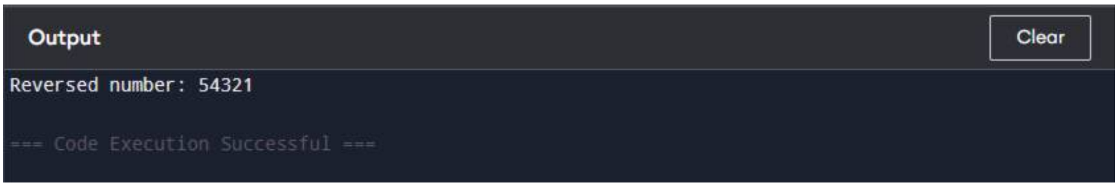


```
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68
    70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100
=== Code Execution Successful ===
```

Q25: Reverse an integer without using string conversion



```java
public class ReverseInteger {
    public static void main(String[] args) {
        int num = 12345, rev = 0;
        while (num != 0) {
            rev = rev * 10 + num % 10;
            num /= 10;
        }
        System.out.println("Reversed number: " + rev);
    }
}
```



```
Reversed number: 54321

=== Code Execution Successful ===
```