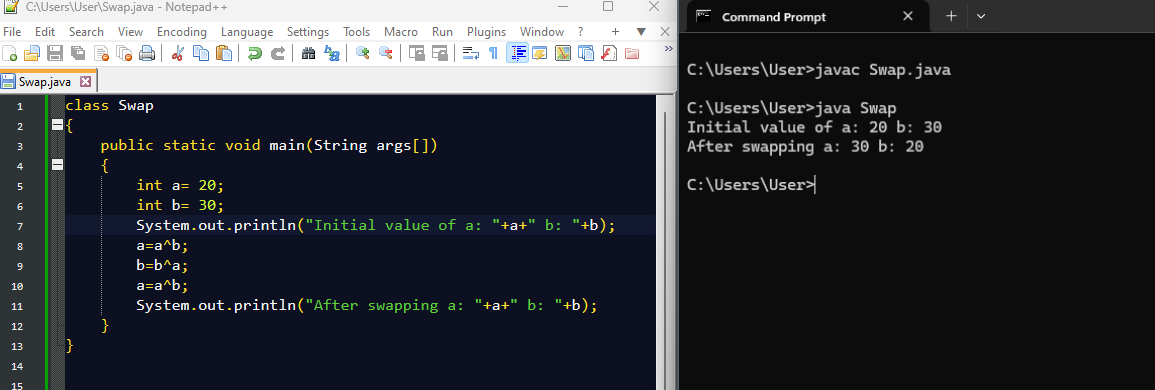
**1. Arithmetic & Assignment Operators**

**Q1: Write a program to swap two numbers without using a third variable and without**

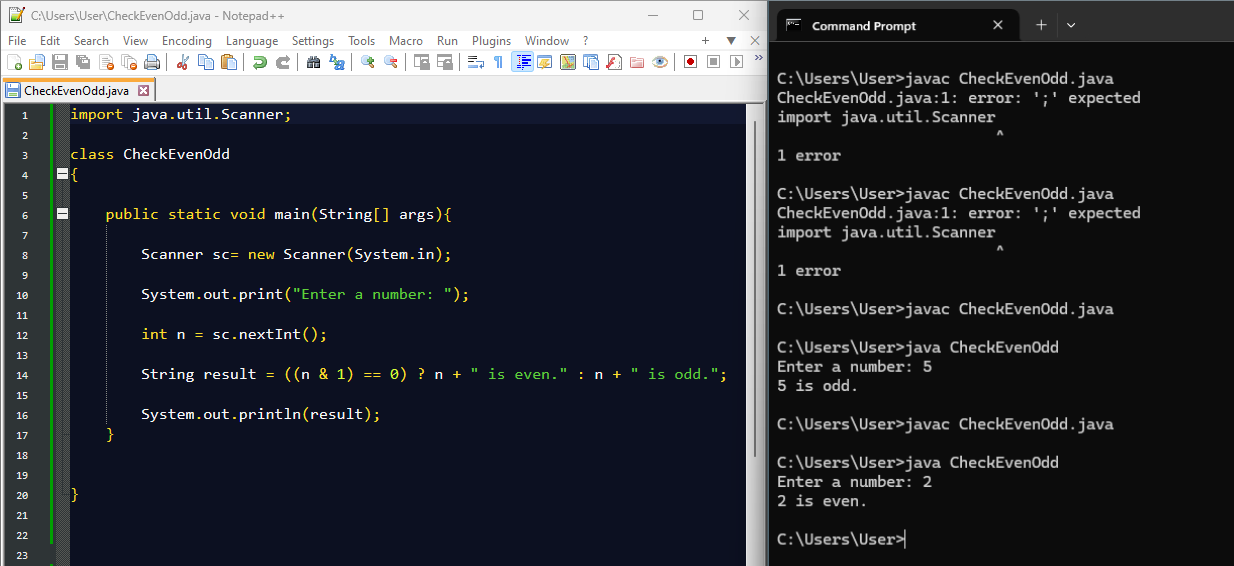
**using arithmetic operators like + or - .**

**Hint : Use bitwise XOR ^ operator.**

****

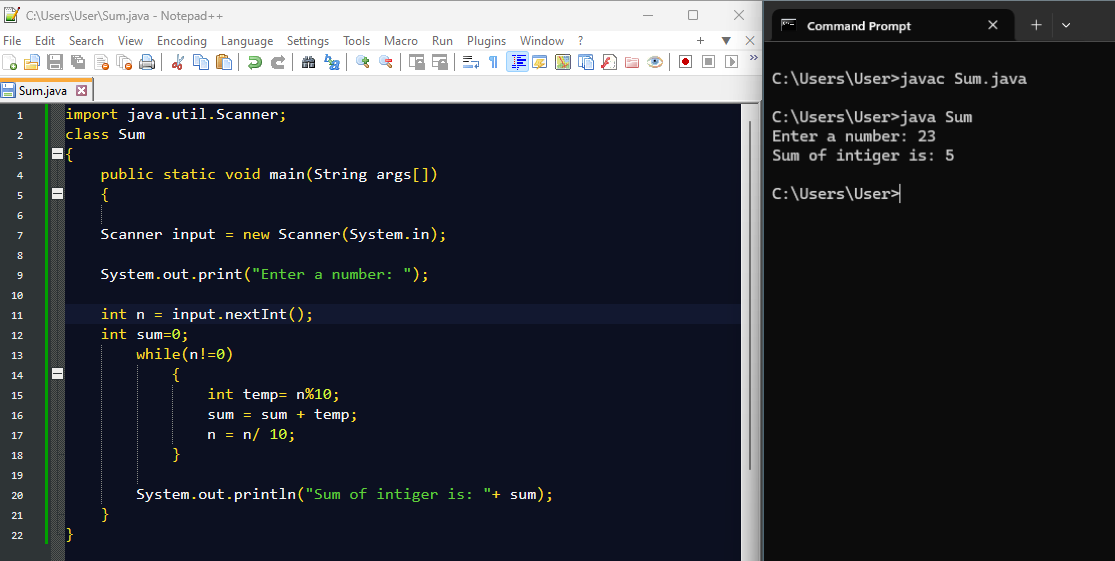
**Q2: Write a program to check whether a given number is even or odd using only bitwise**

**operators .**

**Hint : Use n & 1 to check.**

**Q3: Implement a program that calculates the sum of digits of an integer using modulus**

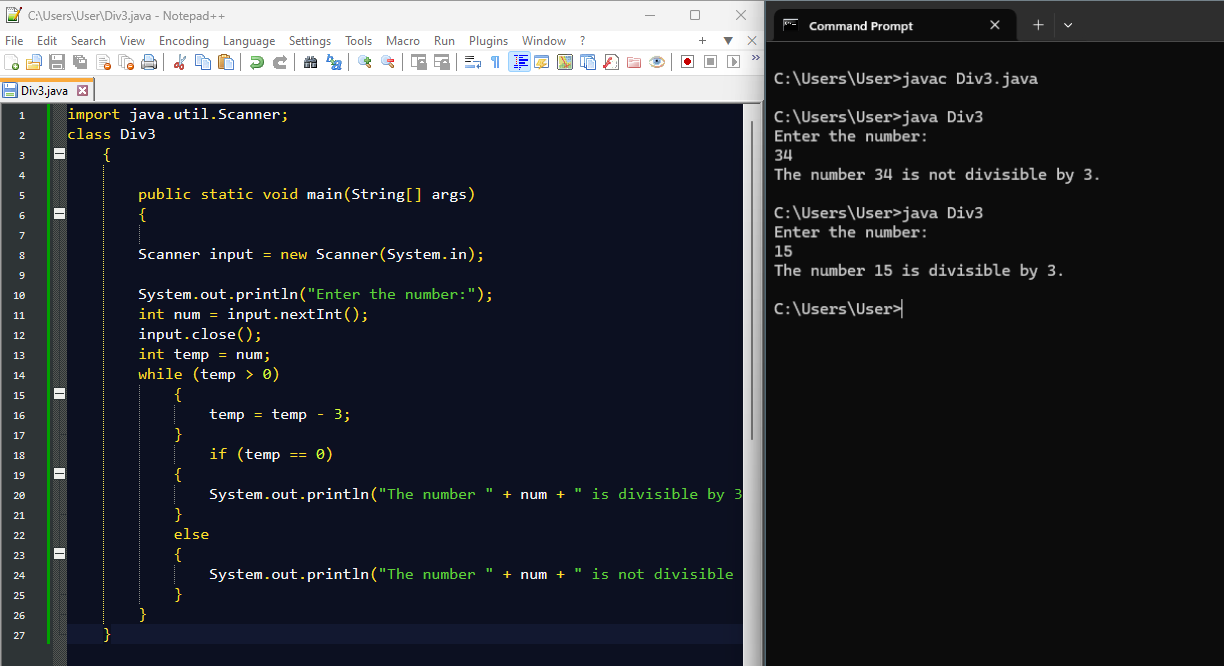
**( % ) and division ( / ) operators .**

****

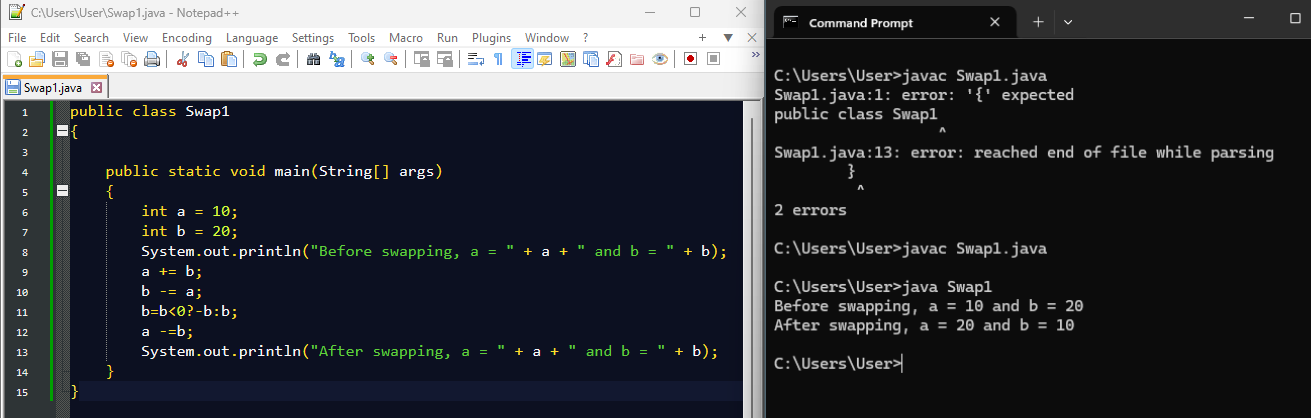
**Q4: Write a program to find whether a given number is divisible by 3 without using the**

**modulus ( % ) or division ( / ) operators.**

**Hint : Use subtraction and bitwise shifts .**

****

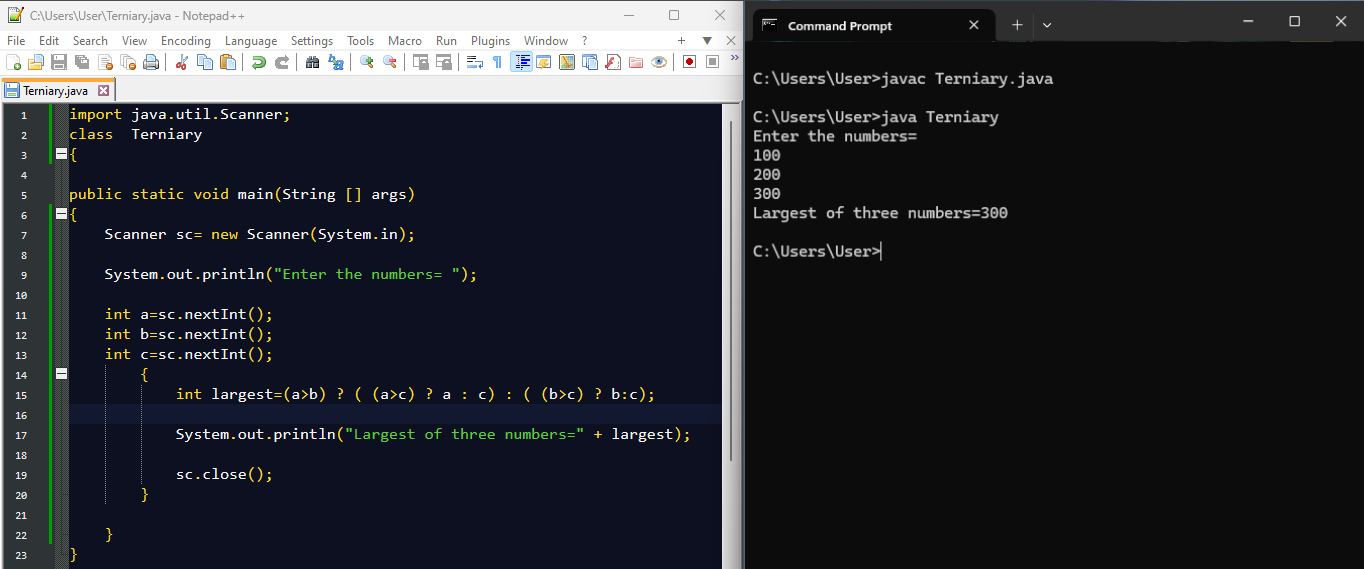
**Q5: Write a Java program to swap two numbers using the += and -= operators**

****

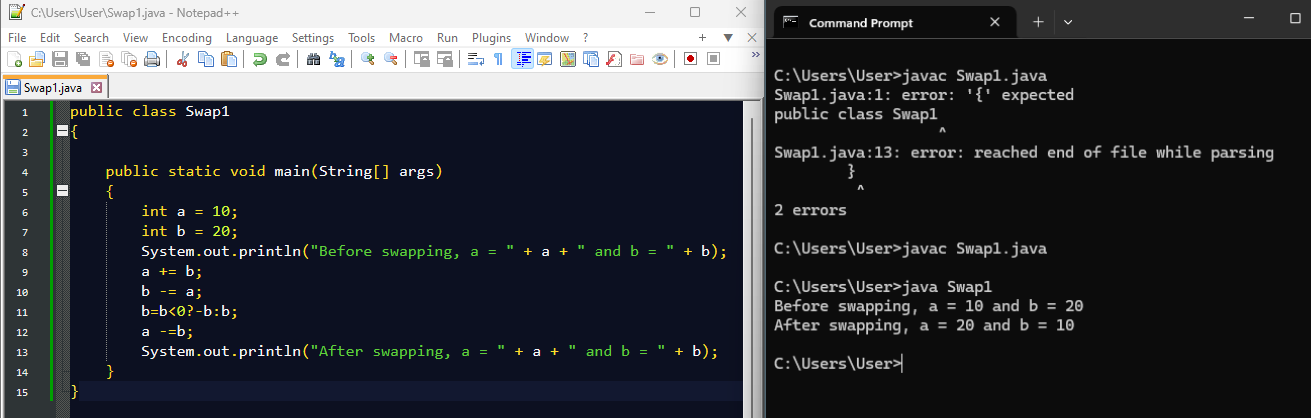
**2. Relational & Logical Operators**

**Q6: Write a program to find the largest of three numbers using only the ternary operator**

**( ? : ) .**

****

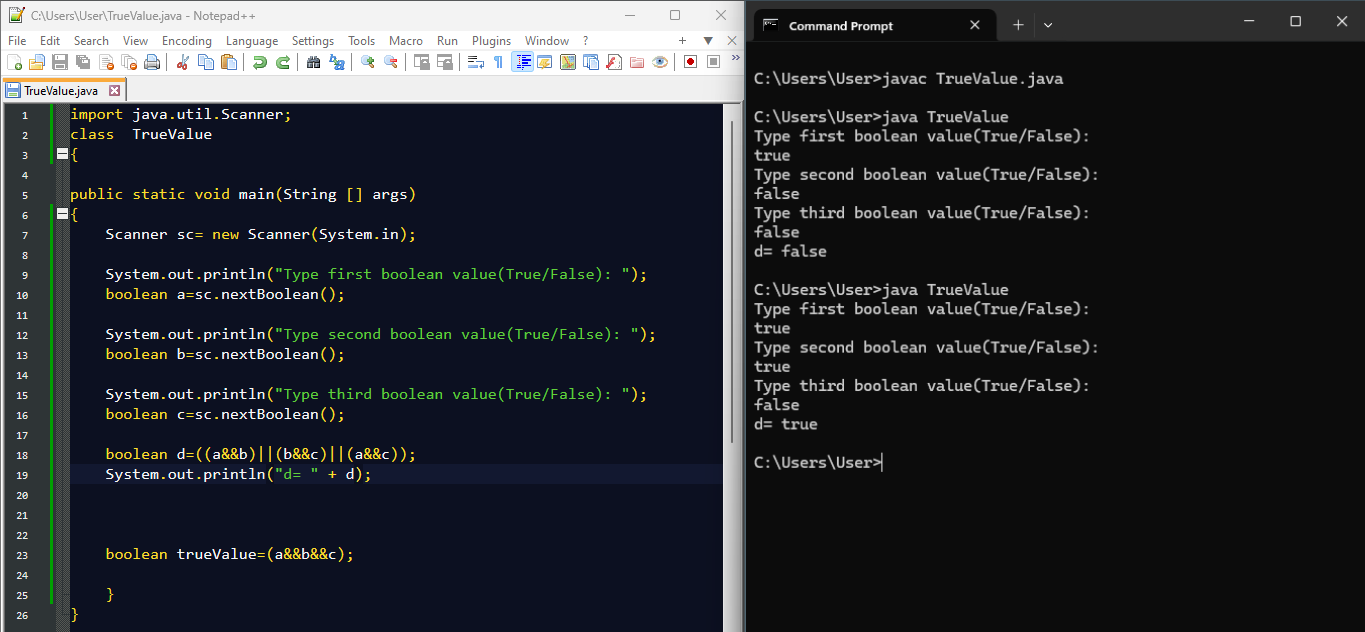
**Q7: Implement a Java program that checks whether a given year is a leap year or not using logical ( && , || ) operators .**

****

**Q8: Write a program that takes three boolean inputs and prints true if at least two of**

**them are true .**

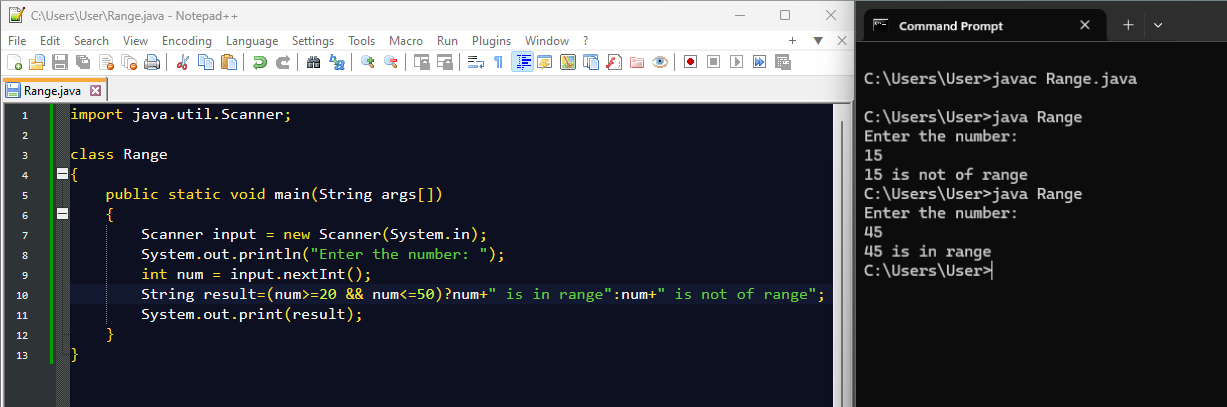
**Hint : Use logical operators ( && , || ).**

****

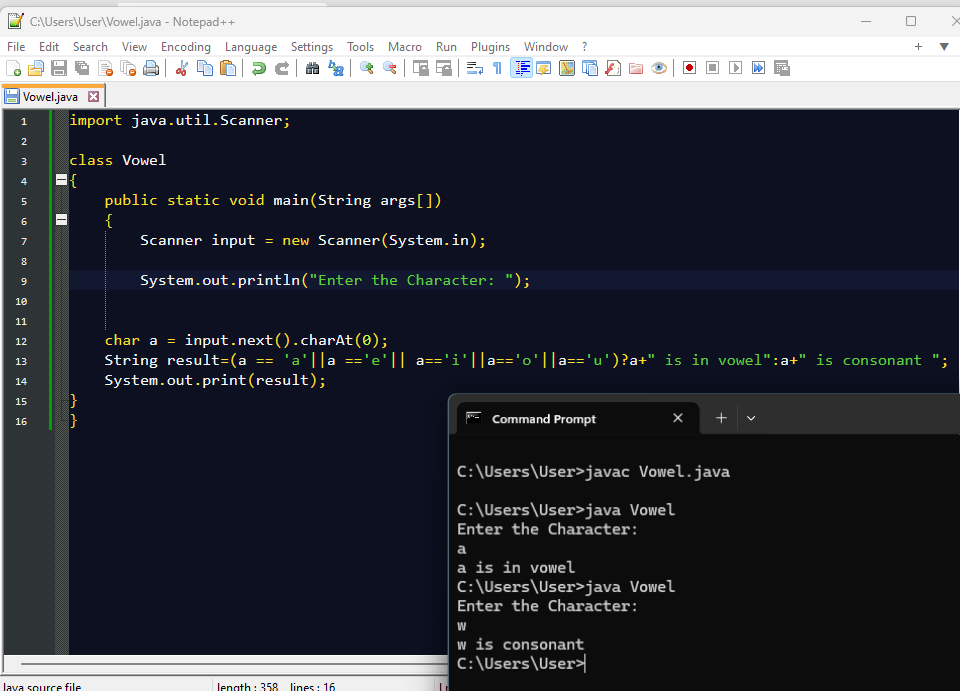
**Q9: Implement a Java program that checks if a number is within a specific range (20 to**

**50) without using if-else .**

**Hint : Use logical AND ( && ) in a print statement .**

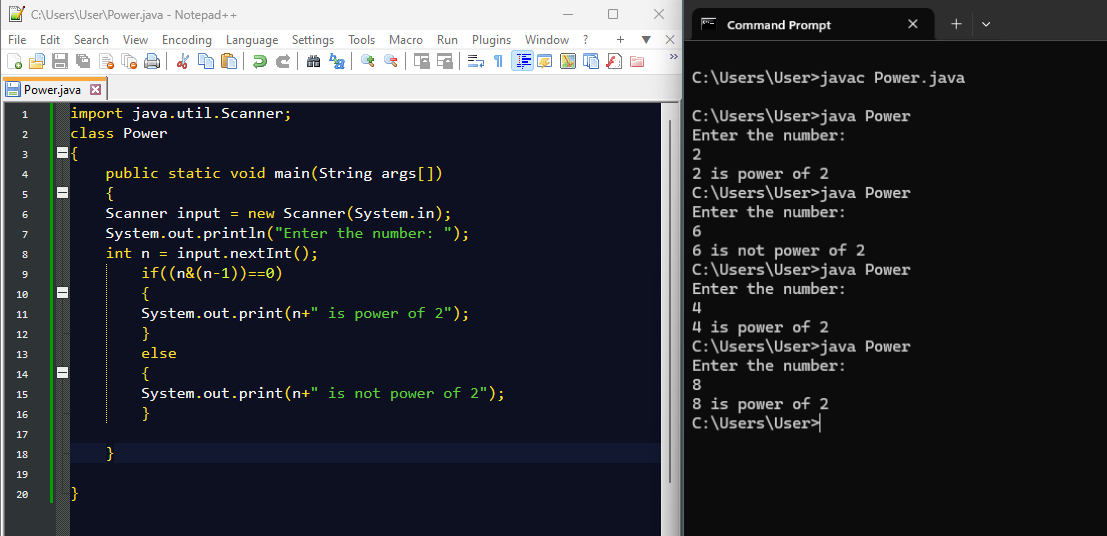
****

**Q10: Write a program to determine if a character is a vowel or a consonant using the ternary operator.**

****

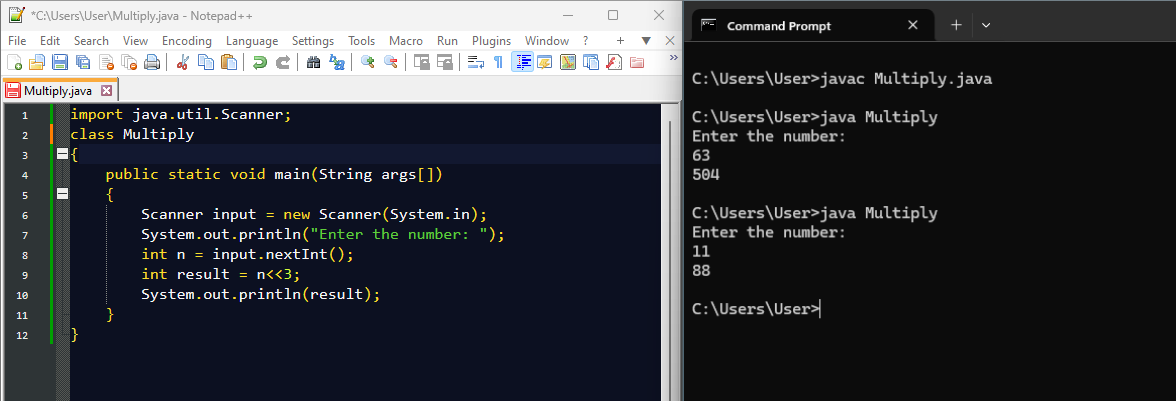
**Q11: Write a program to check if a given number is a power of 2 using bitwise operators.**

**Hint : n & (n - 1) == 0 for positive numbers.**

****

**Q12: Write a Java program to multiply a number by 8 without using \* or / operators.**

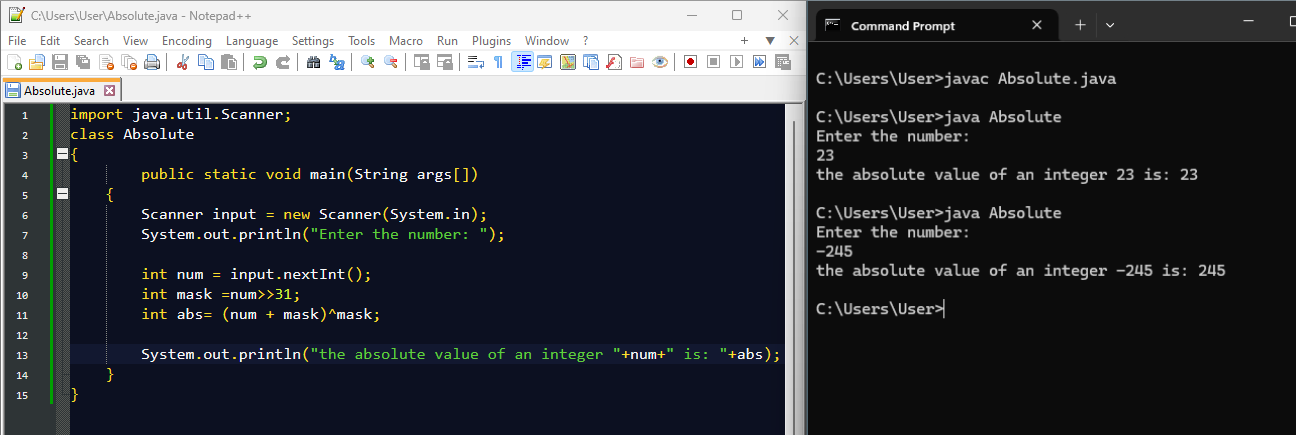
**Hint : Use bitwise left shift ( << ).**

****

**Q13: Implement a Java program to find the absolute value of an integer using bitwise**

**operators.**

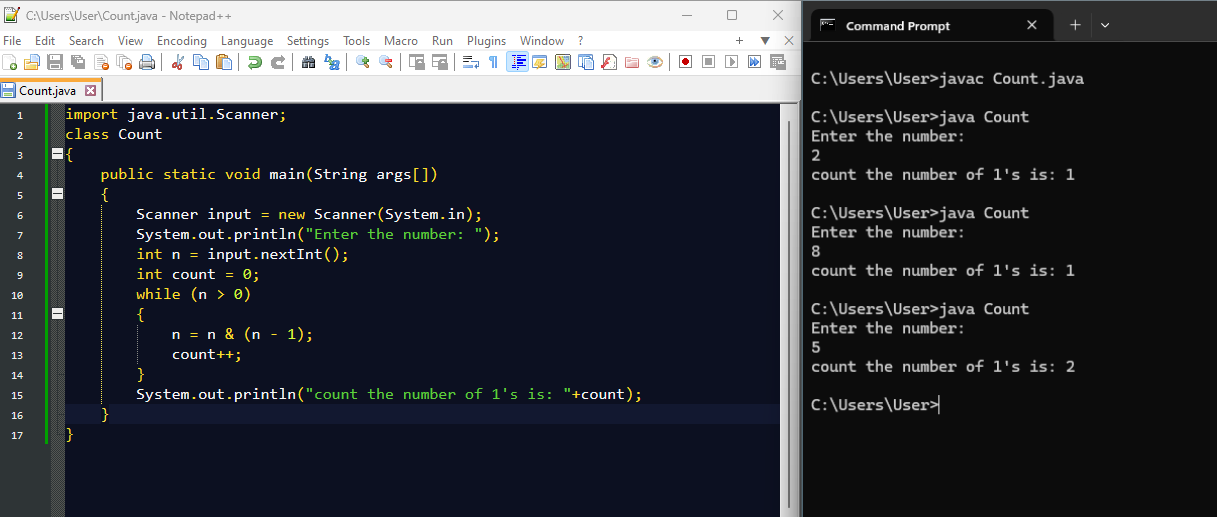
**Hint : mask = num >> 31; abs = (num + mask) ^ mask;**

****

**Q14: Write a program to count the number of 1s (set bits) in a binary representation of a**

**number using bitwise operations.**

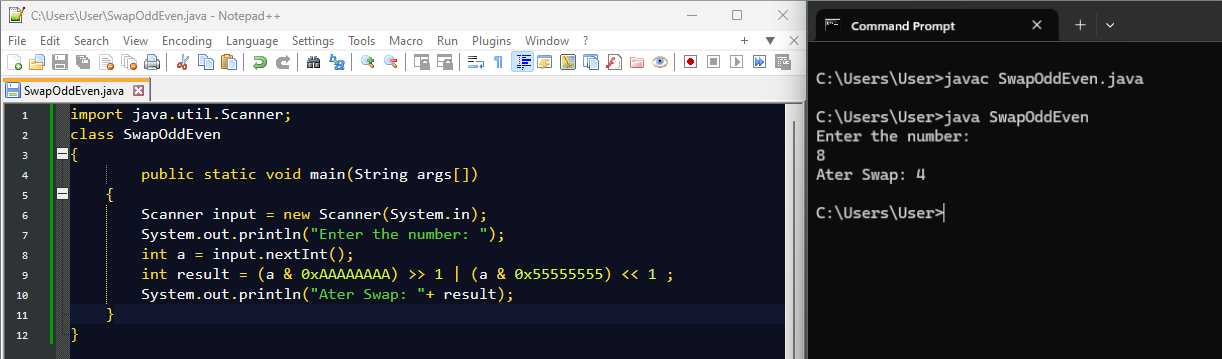
**Hint : Use n & (n - 1) .**

****

**Q15: Implement a program to swap odd and even bits of a number using bitwise**

**operators.**

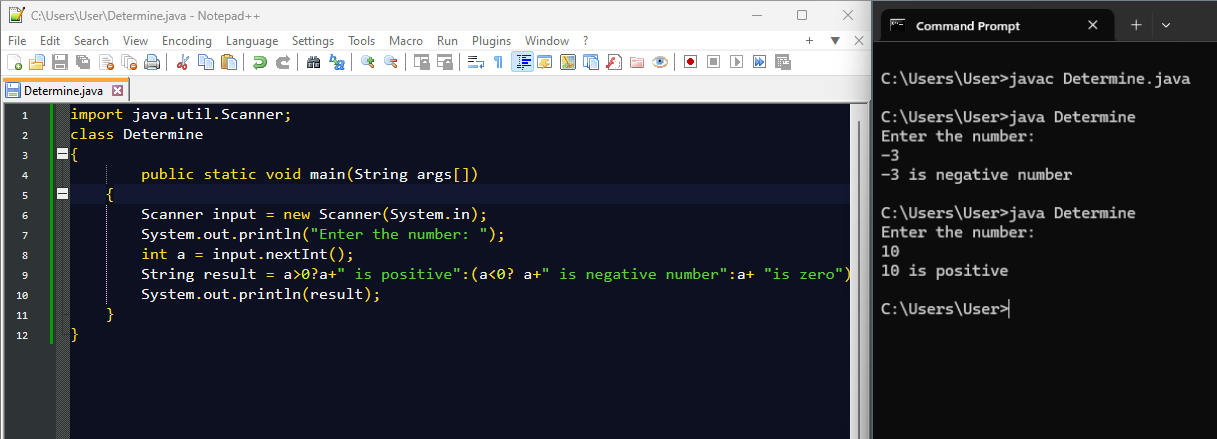
**Hint : Use masks: (x & 0xAAAAAAAA) >> 1 | (x & 0x55555555) << 1**

****

**4. Ternary Operator Challenges**

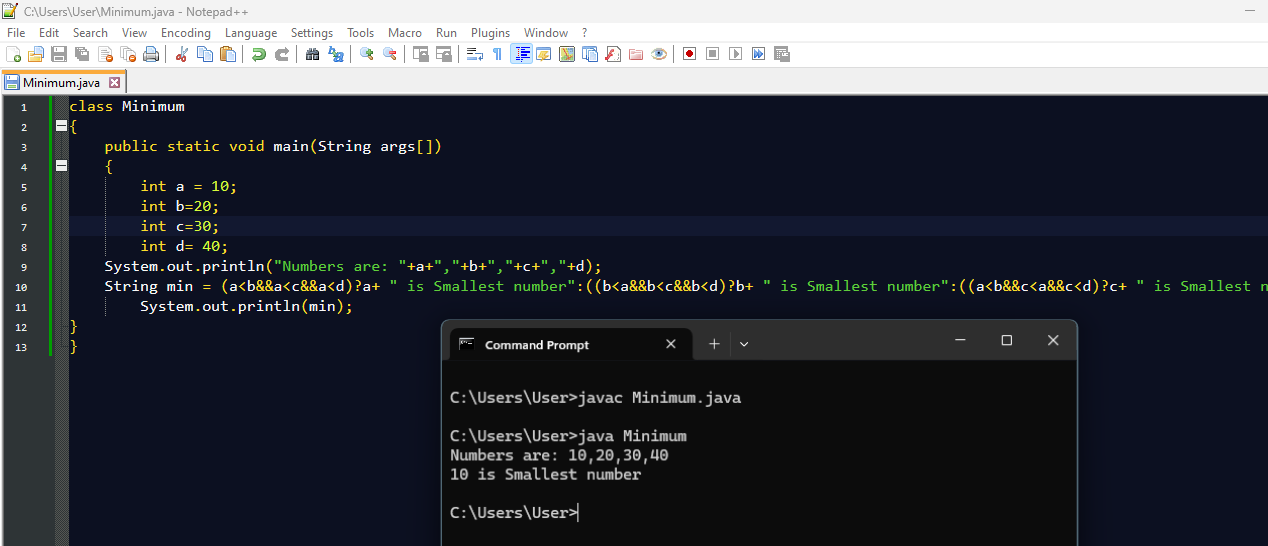
**Q16: Write a program that determines whether a given number is positive, negative, or**

**zero using only the ternary operator .**

****

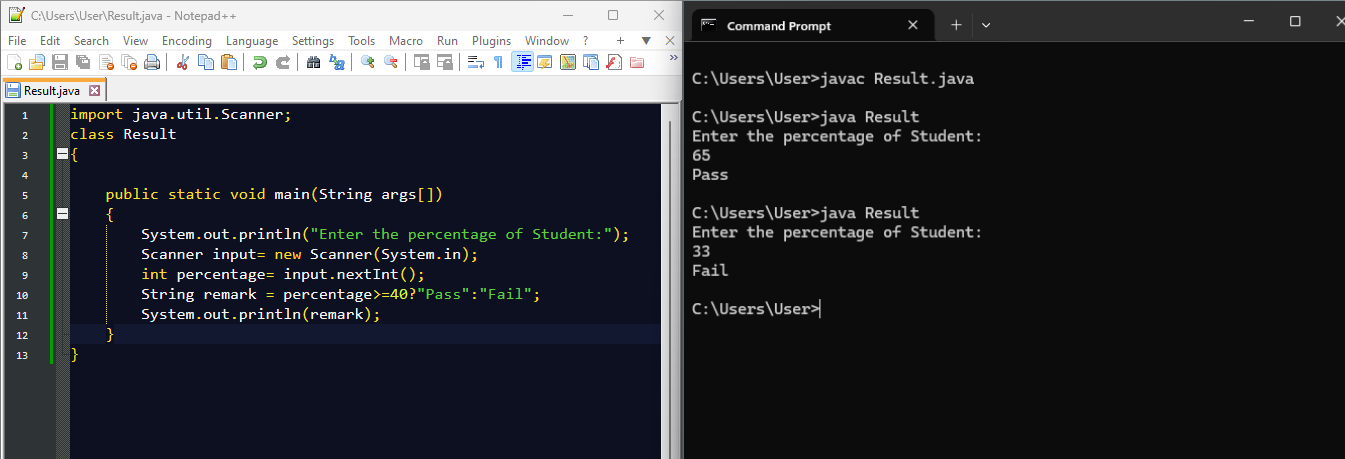
**Q17: Implement a Java program that finds the minimum of four numbers using nested**

**ternary operators.**

****

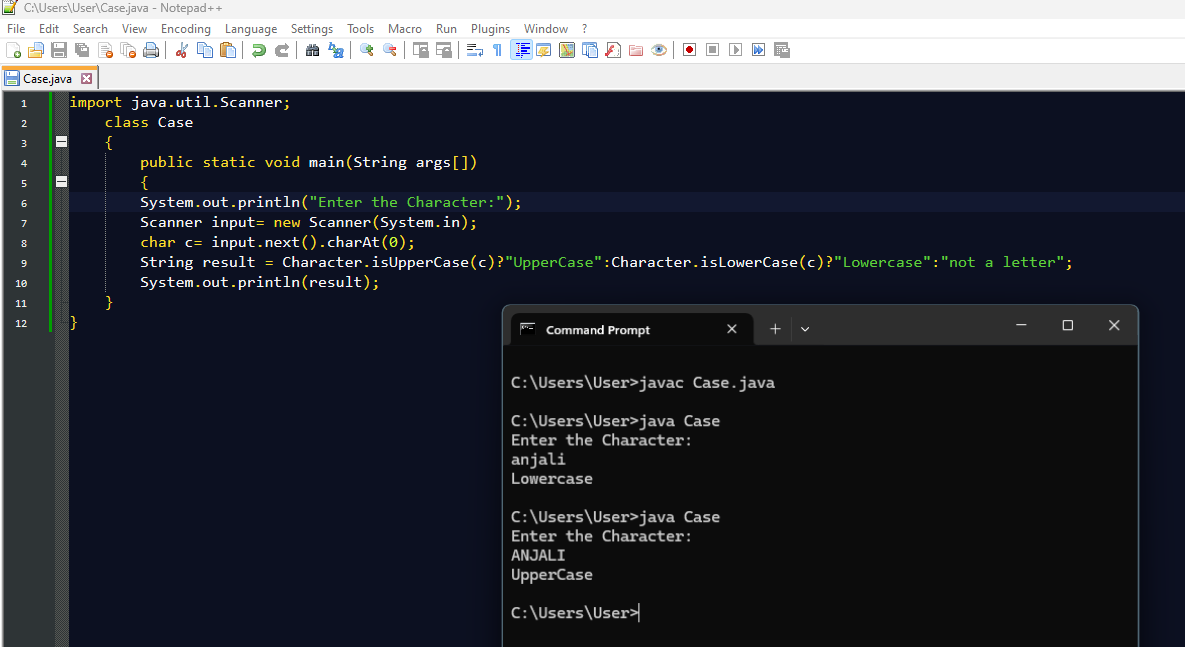
**Q18: Given a student’s percentage, print “Pass” if the percentage is 40 or above;**

**otherwise, print “Fail” , using only the ternary operator.**

****

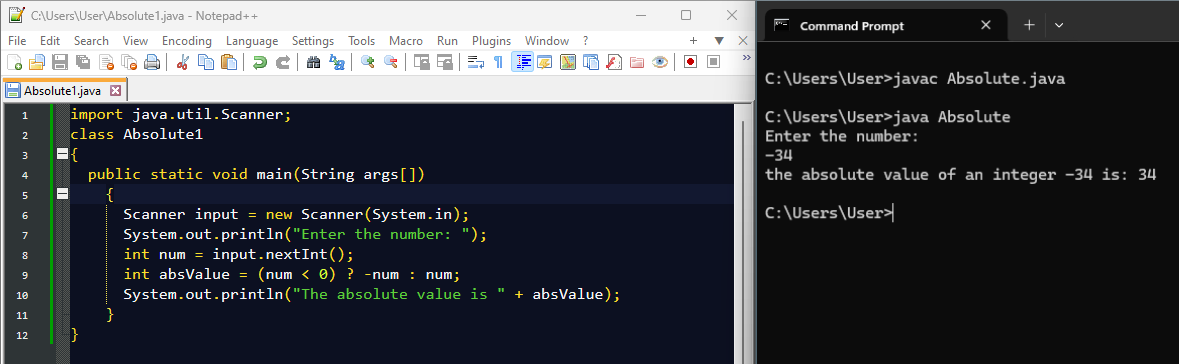
**Q19: Write a Java program that checks whether a character is uppercase, lowercase, or**

**not a letter using only the ternary operator.**

****

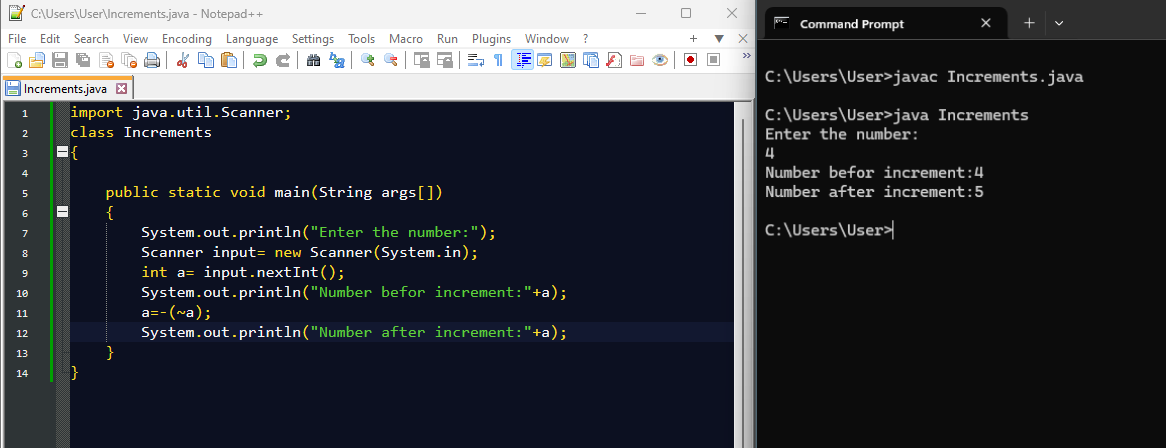
**Q20: Implement a Java program that returns the absolute value of a given number using**

**the ternary operator (without using Math.abs() ).**

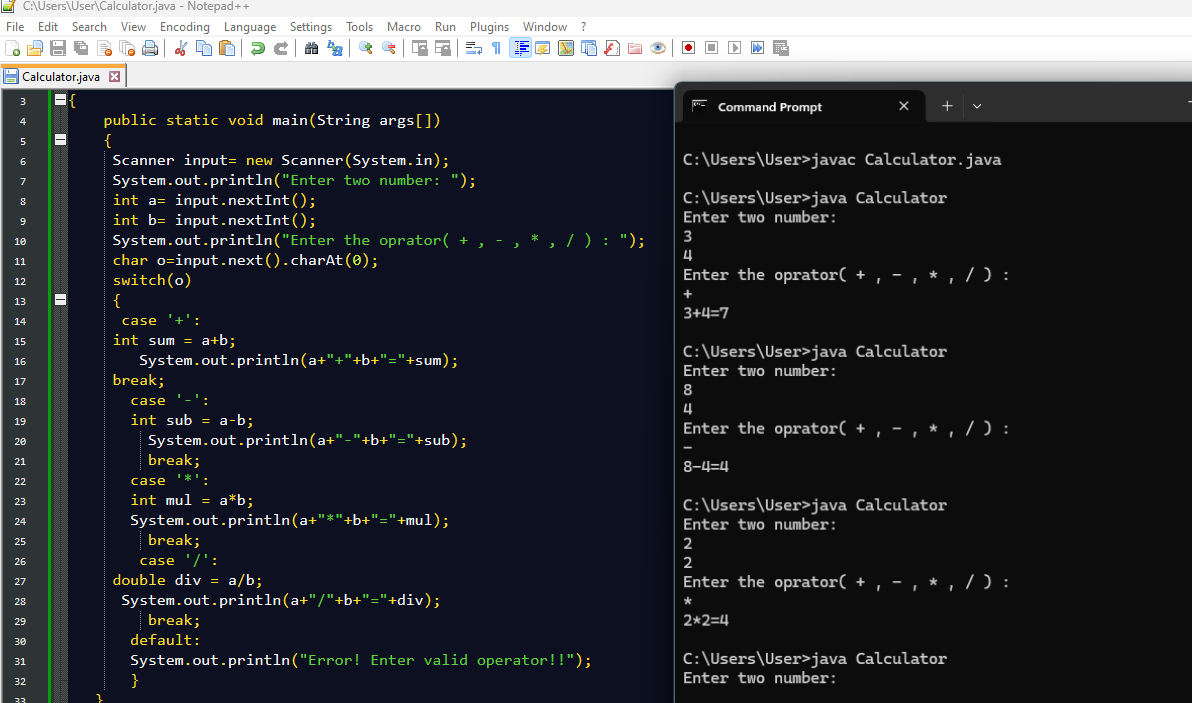
****

**Q21: Write a program that increments a number without using + or ++ operators.**

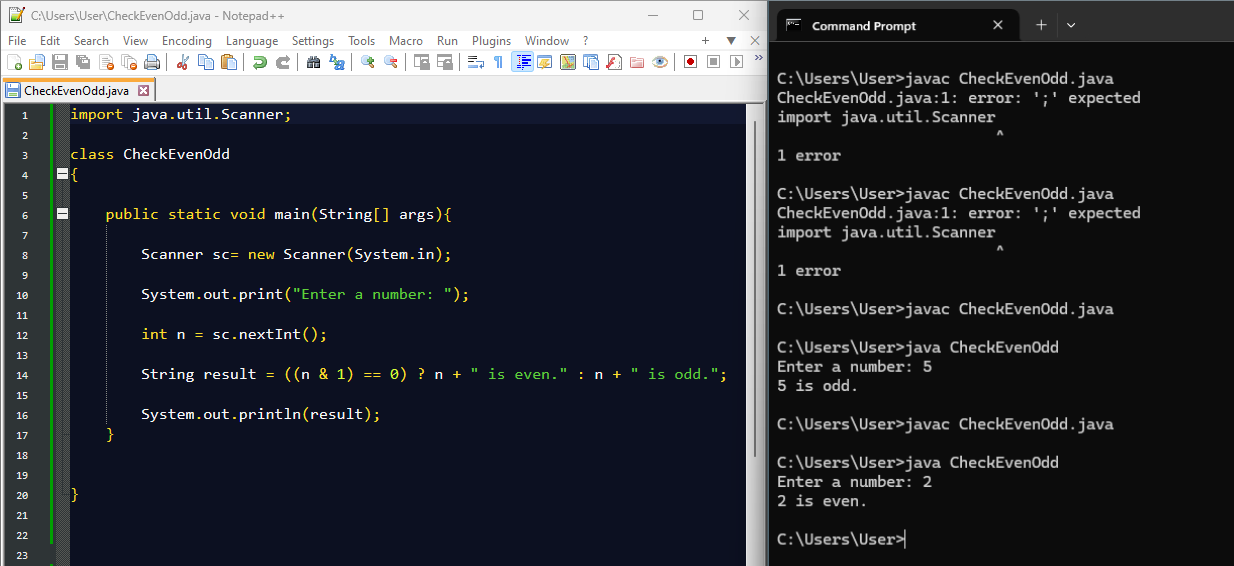
**Hint : Use bitwise - (~x) .**

****

**Q22: Implement a calculator that takes two numbers and an operator ( + , - , \* , / ) as input and prints the result using only switch-case .**

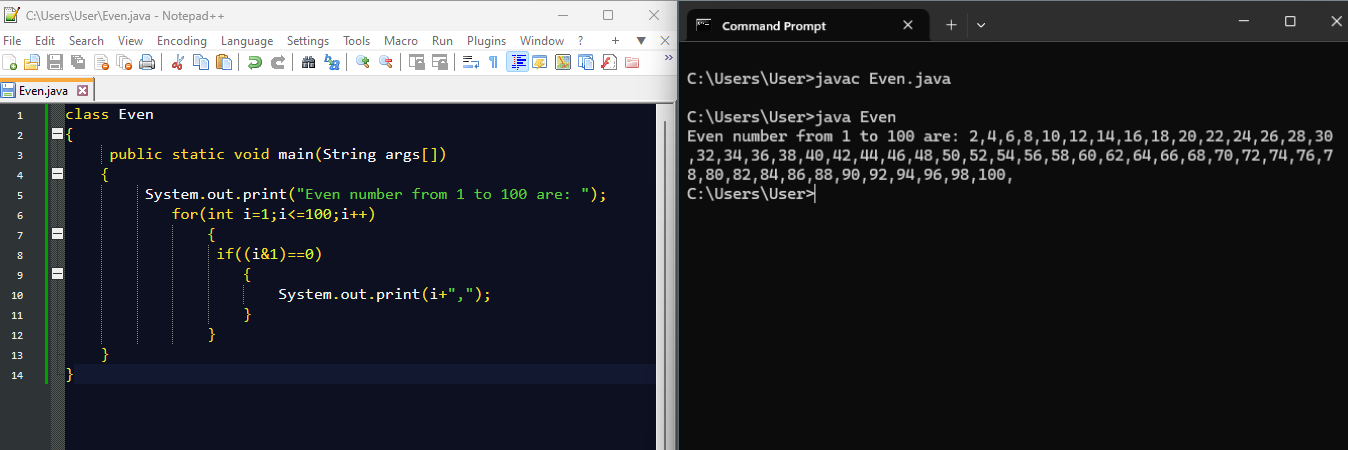
****

**Q23: Given a number, find whether it is odd or even using the & bitwise operator and print the result without using if-else .**

****

**Q24: Write a program that prints all even numbers from 1 to 100 using only bitwise AND**

**( & ) and for loop.**

****

**Q25: Implement a program that reverses an integer number without using string**

**conversion ( StringBuilder or toCharArray ).**

**Hint : Use while(n!=0) { rev = rev \* 10 + n % 10; n /= 10; }**

