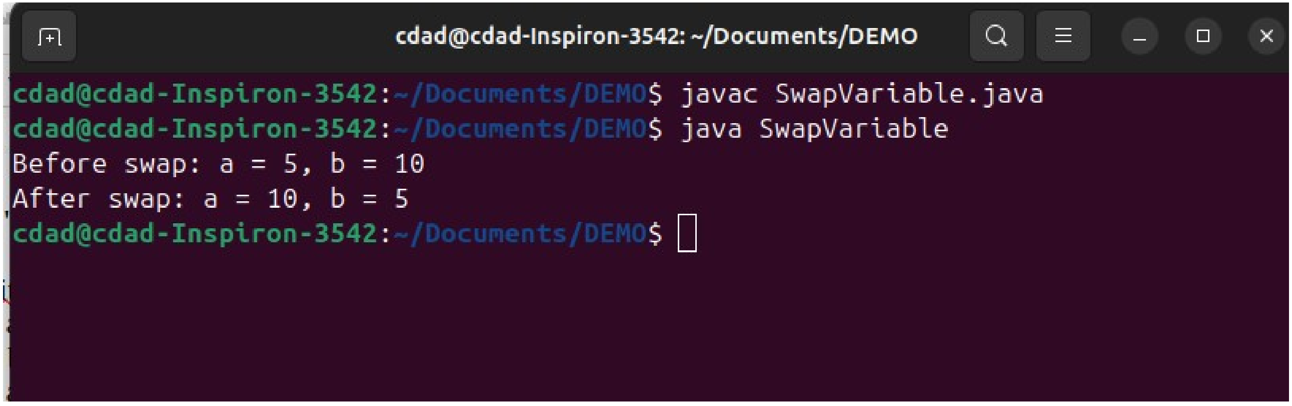
ASSIGNMENT NO : 2

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.

Ans.

```
public class SwapVariable {
    public static void main(String[] args) {
        int a = 5;
        int b = 10;

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

        // Swapping with bitwise XOR
        a = a ^ b; // Step 1: a now contains the XOR of a and b
        b = a ^ b; // Step 2: b now contains the original value of a
        a = a ^ b; // Step 3: a now contains the original value of b

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

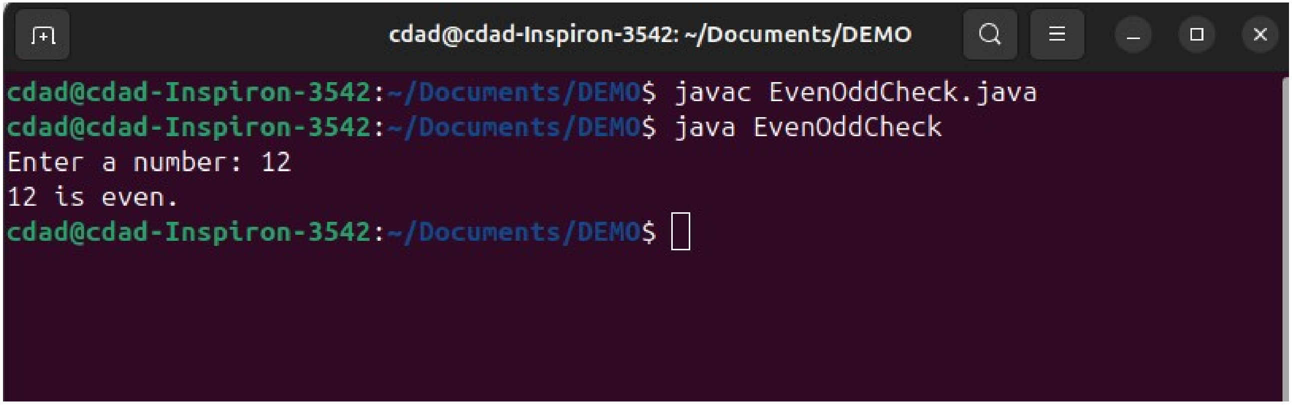


```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac SwapVariable.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java SwapVariable
Before swap: a = 5, b = 10
After swap: a = 10, b = 5
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

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

Ans.

```
import java.util.Scanner;

public class EvenOddCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int n = scanner.nextInt();

        // Using bitwise AND operator to check even or odd
        if ((n & 1) == 0) {
            System.out.println(n + " is even.");
        } else {
            System.out.println(n + " is odd.");
        }

    }
}
```

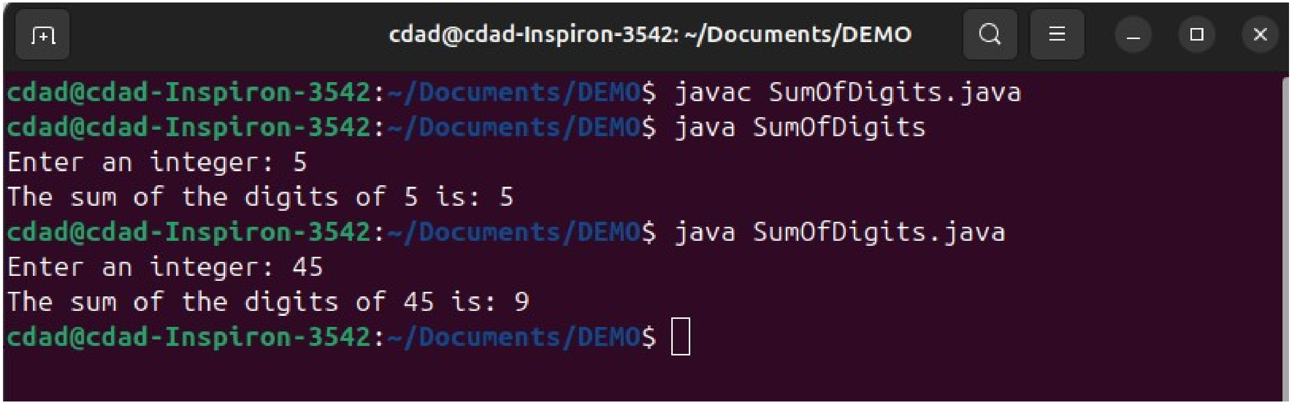


```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac EvenOddCheck.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java EvenOddCheck
Enter a number: 12
12 is even.
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

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

Ans.

```
import java.util.Scanner;

public class SumOfDigits {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();
        // Calculate the sum of the digits
        int sum = sumOfDigits(number);

        // Output the result
        System.out.println("The sum of the digits of " + number + " is: " + sum);

        scanner.close();
    }

    public static int sumOfDigits(int number) {
          int sum = 0;

        number = Math.abs(number);

        while (number > 0) {

            sum += number % 10;

            number /= 10;
        }

        return sum;
}
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac SumOfDigits.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java SumOfDigits
Enter an integer: 5
The sum of the digits of 5 is: 5
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java SumOfDigits.java
Enter an integer: 45
The sum of the digits of 45 is: 9
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

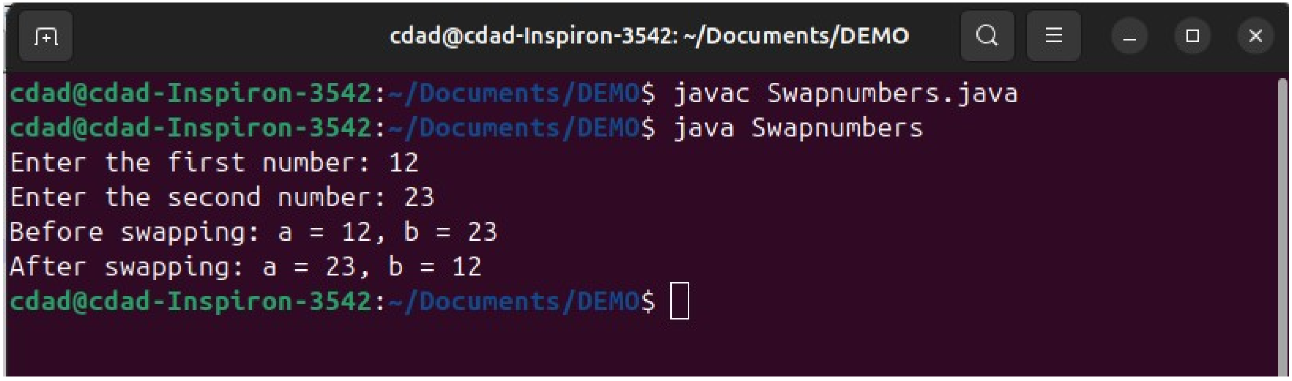
Q4. Write a program to find whether a given number is divisible by 3 without using the modulus(%) or division (/) operators.
Hint: Use subtraction and bitwise shifts.

Ans.

```
import java.util.Scanner;

public class DivisibleBy3{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = scanner.nextInt();

        if (isDivisibleBy3(n)) {
            System.out.println(n + " is divisible by 3.");
        } else {
            System.out.println(n + " is not divisible by 3.");
        }

        scanner.close();
    }
           public static boolean isDivisibleBy3(int n) {

        if (n < 0) {
            n = -n;
        }

        while (n > 3) {
            n -= 3; // Subtract 3
        }

        return n == 0; // if number is exactly 0, it's divisible by 3
    }
}
```

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

```
import java.util.Scanner;

public class Swapnumbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int a = scanner.nextInt();
        System.out.print("Enter the second number: ");
        int b = scanner.nextInt();

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

        // Swap numbers using += and -=
        a += b; // Now a contains the sum of both numbers
        b = a - b; // b becomes the original value of a
        a -= b; // a becomes the original value of b

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

        scanner.close();
    }
}
```

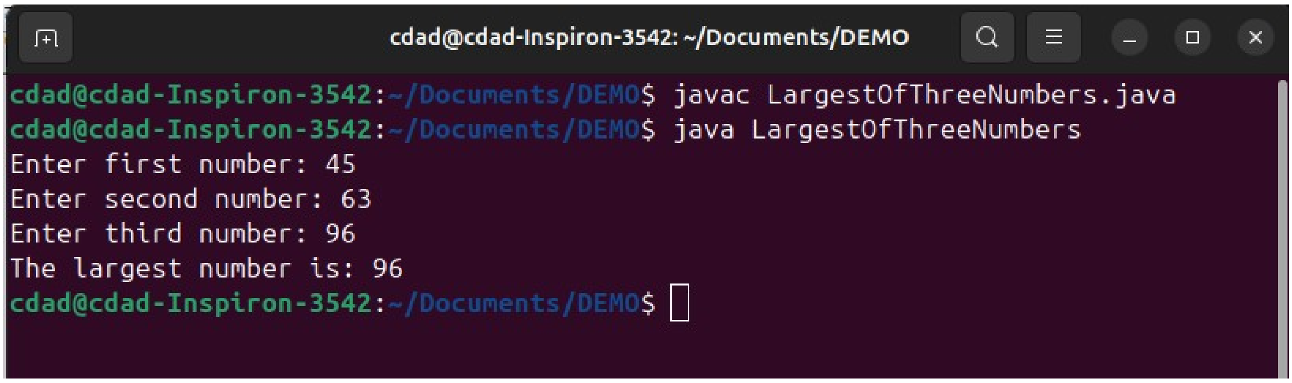


```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac Swapnumbers.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java Swapnumbers
Enter the first number: 12
Enter the second number: 23
Before swapping: a = 12, b = 23
After swapping: a = 23, b = 12
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

2. Relational & Logical Operators.

Q6. Write a program to find the largest of three numbers using only the ternary operators.(? :).
Ans.

```
import java.util.Scanner;

public class LargestOfThreeNumbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first number: ");
        int num1 = scanner.nextInt();

        System.out.print("Enter second number: ");
        int num2 = scanner.nextInt();

        System.out.print("Enter third number: ");
        int num3 = scanner.nextInt();

        int largest = (num1 > num2) ? ((num1 > num3) ? num1 : num3) : ((num2 >
num3) ? num2 : num3);
        System.out.println("The largest number is: " + largest);
        scanner.close();
    }
}
```

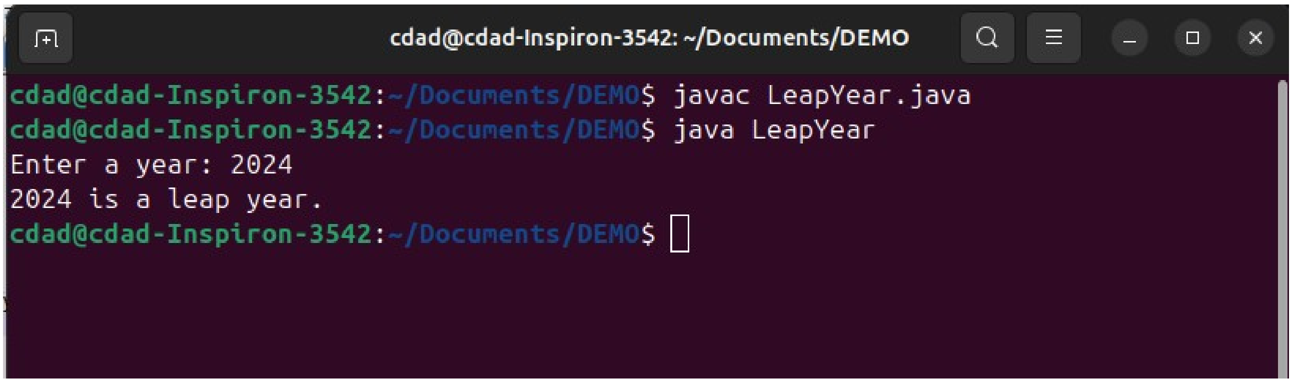


```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac LargestOfThreeNumbers.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java LargestOfThreeNumbers
Enter first number: 45
Enter second number: 63
Enter third number: 96
The largest number is: 96
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

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

Ans.

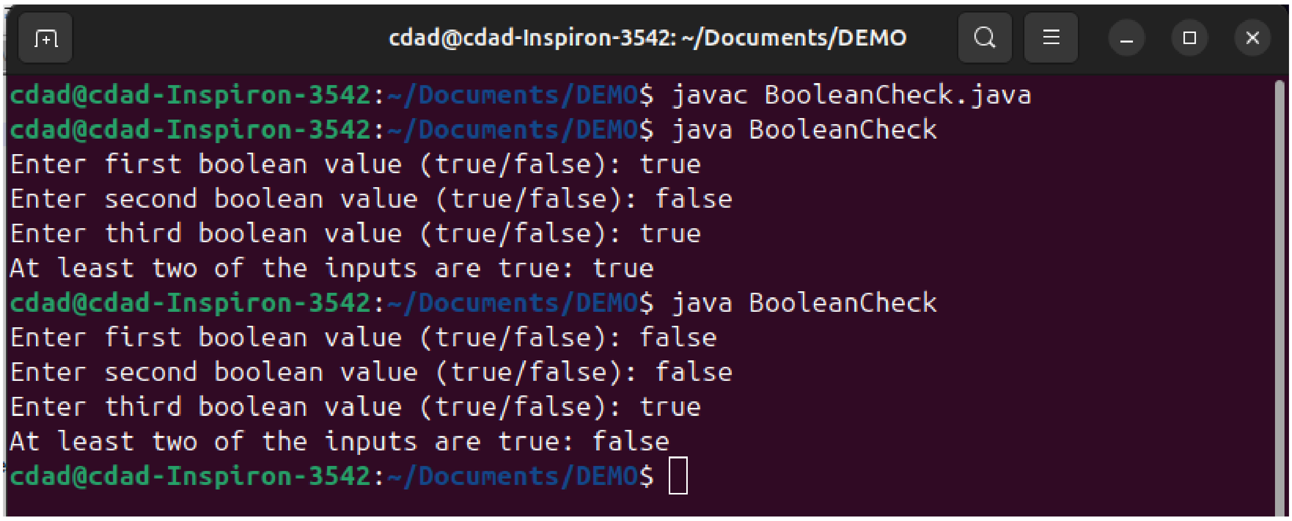
```
import java.util.Scanner;

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

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a year: ");
        int year = scanner.nextInt();

        boolean isLeapYear = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

        if (isLeapYear) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }
        scanner.close();
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac LeapYear.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java LeapYear
Enter a year: 2024
2024 is a leap year.
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

Q8. Write a program that takes three boolean inputs and prints true if at least two of them are true.
Hint : Use logical operators (&&, ||).
Ans.

```
import java.util.Scanner;

public class BooleanCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first boolean value (true/false): ");
        boolean a = scanner.nextBoolean();

        System.out.print("Enter second boolean value (true/false): ");
        boolean b = scanner.nextBoolean();

        System.out.print("Enter third boolean value (true/false): ");
        boolean c = scanner.nextBoolean();

        boolean atLeastTwoTrue = (a && b) || (a && c) || (b && c);
        System.out.println("At least two of the inputs are true: " + atLeastTwoTrue);

        scanner.close();
    }
}
```

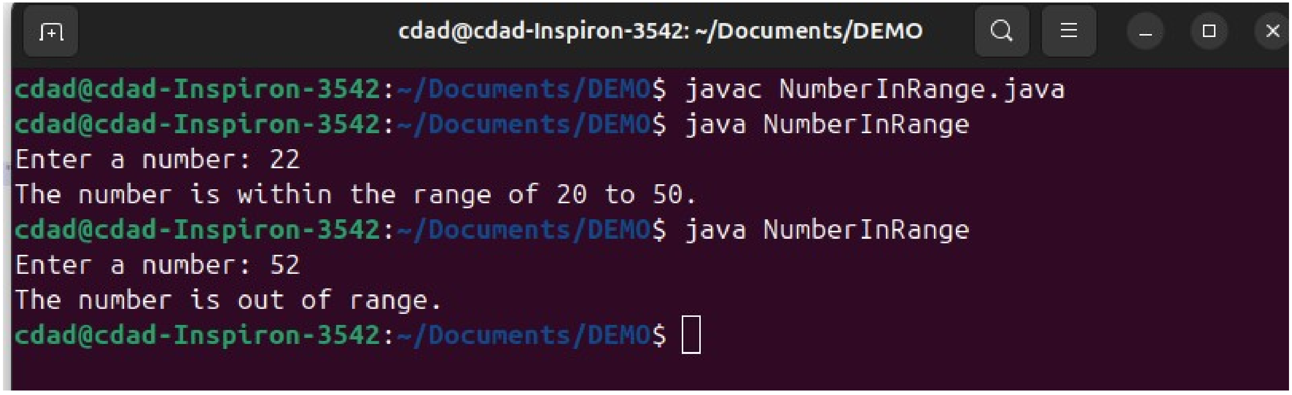


```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac BooleanCheck.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java BooleanCheck
Enter first boolean value (true/false): true
Enter second boolean value (true/false): false
Enter third boolean value (true/false): true
At least two of the inputs are true: true
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java BooleanCheck
Enter first boolean value (true/false): false
Enter second boolean value (true/false): false
Enter third boolean value (true/false): true
At least two of the inputs are true: false
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

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.

Ans.

```
import java.util.Scanner;

public class NumberInRange {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

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

        System.out.println((number >= 20 && number <= 50) ? "The number is within the range of 20 to 50." : "The number is out of range.");

        scanner.close();
    }
}
```

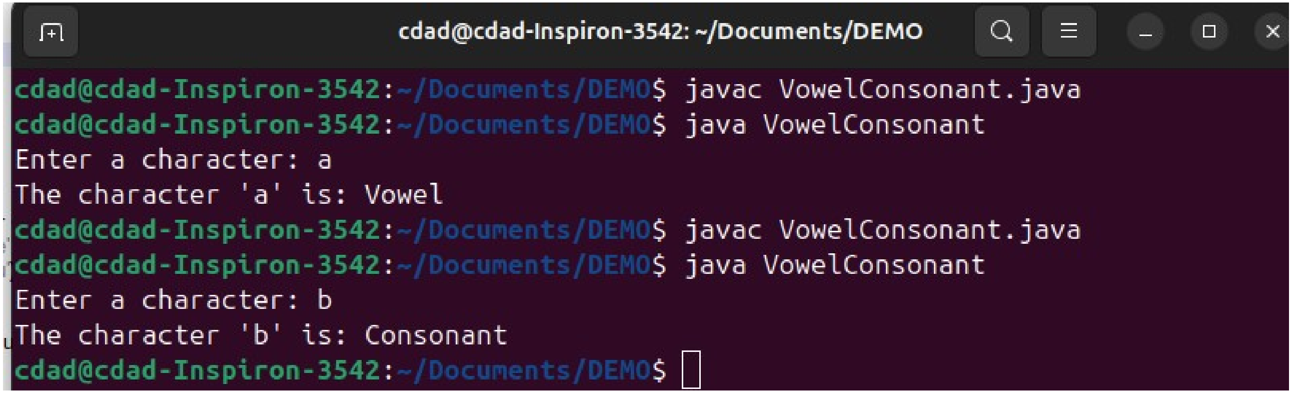


```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac NumberInRange.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java NumberInRange
Enter a number: 22
The number is within the range of 20 to 50.
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java NumberInRange
Enter a number: 52
The number is out of range.
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

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

Ans.

```
import java.util.Scanner;

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

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a character: ");
        char ch = scanner.next().charAt(0);

        char lowercaseChar = Character.toLowerCase(ch);

        String result = (lowercaseChar >= 'a' && lowercaseChar <= 'z') ?
                        (lowercaseChar == 'a' || lowercaseChar == 'e' || lowercaseChar == 'i' ||
                         lowercaseChar == 'o' || lowercaseChar == 'u') ? "Vowel" :
"Consonant"
                        : "Not an alphabetic character";

        System.out.println("The character '" + ch + "' is: " + result);
    }
          }
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac VowelConsonant.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java VowelConsonant
Enter a character: a
The character 'a' is: Vowel
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac VowelConsonant.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java VowelConsonant
Enter a character: b
The character 'b' is: Consonant
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

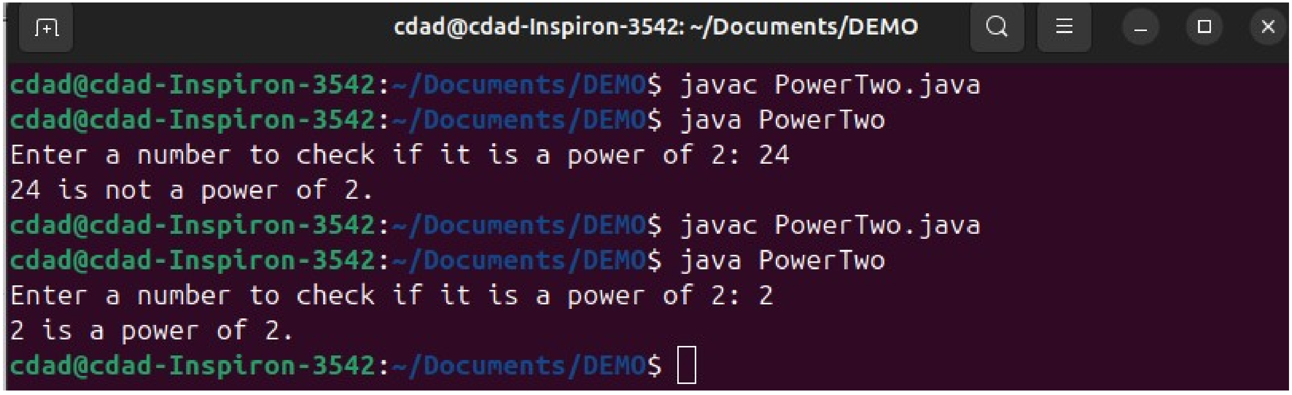
3. Bitwise Operators

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.

Ans.

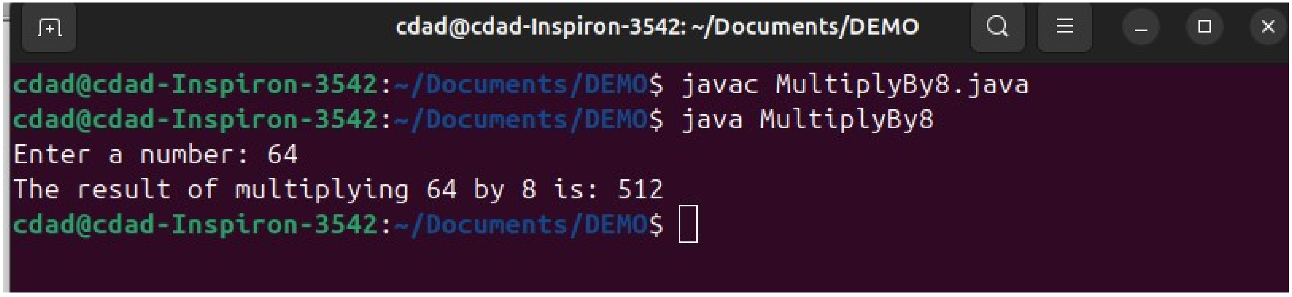
```
import java.util.Scanner;

public class PowerTwo{

    public static boolean isPowerTwo(int n) {
        return n > 0 && (n & (n - 1)) == 0;
    }

      public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number to check if it is a power of 2: ");
        int number = scanner.nextInt();

        if (isPowerTwo(number)) {
            System.out.println(number + " is a power of 2.");
        } else {
            System.out.println(number + " is not a power of 2.");
        }

        scanner.close();
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac PowerTwo.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java PowerTwo
Enter a number to check if it is a power of 2: 24
24 is not a power of 2.
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac PowerTwo.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java PowerTwo
Enter a number to check if it is a power of 2: 2
2 is a power of 2.
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

Q12:Write a Java program to multiply a number by 8 without using *or/operators.
Hint: Use bitwise left shift (<<).
Ans.

```
import java.util.Scanner;

public class MultiplyBy8 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

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

        int result = number << 3;

        System.out.println("The result of multiplying " + number + " by 8 is: " + result);

        scanner.close();
    }
}
```



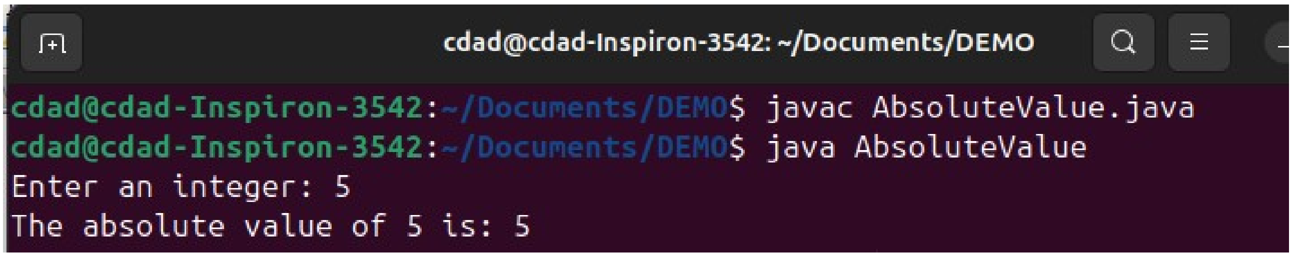
```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac MultiplyBy8.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java MultiplyBy8
Enter a number: 64
The result of multiplying 64 by 8 is: 512
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

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

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

Ans.

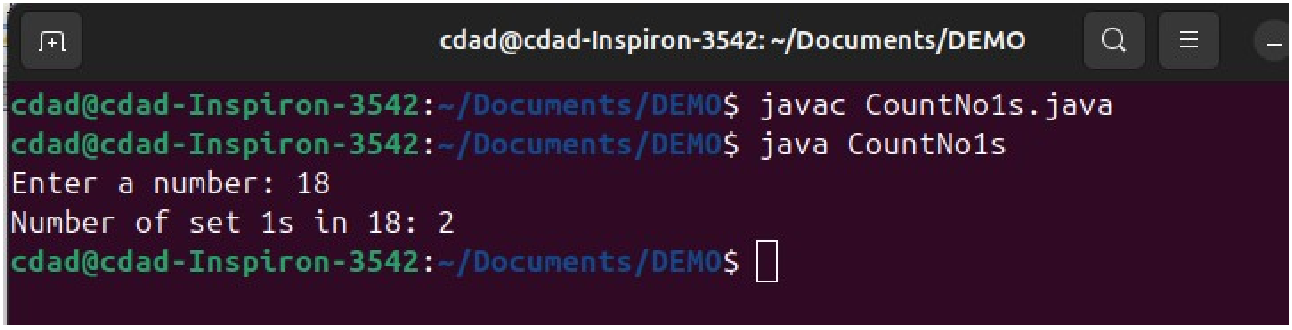
```
import java.util.Scanner;

public class AbsoluteValue {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int num = scanner.nextInt();

        int absValue = absoluteValue(num);
        System.out.println("The absolute value of " + num + " is: " + absValue);
    }

    public static int absoluteValue(int num) {
        int mask = num >> 31; // Shift the sign bit to the rightmost position
        return (num + mask) ^ mask; // Apply the formula
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac AbsoluteValue.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java AbsoluteValue
Enter an integer: 5
The absolute value of 5 is: 5
```

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)
Ans.

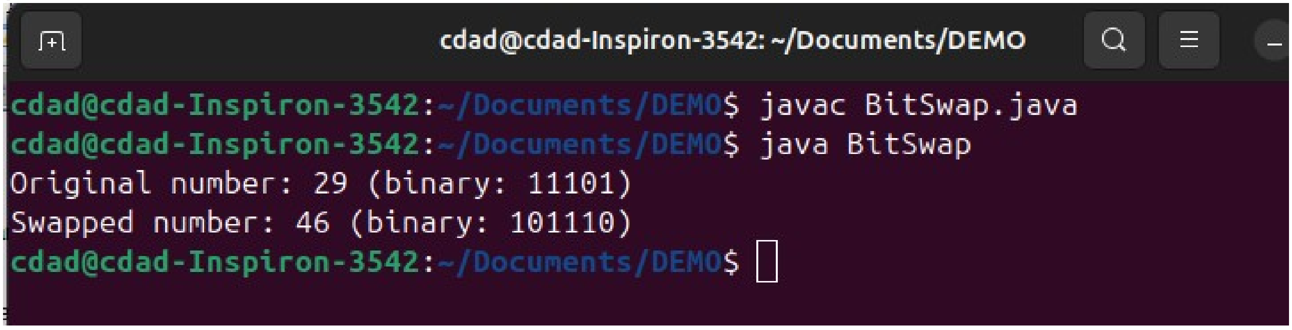
```
import java.util.Scanner;

public class CountNo1s {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

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

        int count = countSetBits(number);

        System.out.println("Number of set 1s in " + number + ": " + count);
    }

    public static int countSetBits(int n) {
        int count = 0;
        while (n > 0) {
            n = n & (n - 1);
            count++;        // Increment the count of set bits
        }
        return count;
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac CountNo1s.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java CountNo1s
Enter a number: 18
Number of set 1s in 18: 2
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

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
Ans.

```
public class BitSwap{
    public static void main(String[] args) {
        int number = 29; // Example number (binary: 11101)
        System.out.println("Original number: " + number + " (binary: " + 
Integer.toBinaryString(number) + ")");

        int swappedNumber = swapOddEvenBits(number);
        System.out.println("Swapped number: " + swappedNumber + " (binary: " + 
Integer.toBinaryString(swappedNumber) + ")");
    }

    public static int swapOddEvenBits(int x) {
        // Masks for odd and even bits
        int evenMask = 0xAAAAAAAA; // Mask for even bits
        int oddMask = 0x55555555;  // Mask for odd bits

        // Shift even bits right and odd bits left
        int evenBits = (x & evenMask) >> 1;
        int oddBits = (x & oddMask) << 1;
        // Combine even and odd bits
        return evenBits | oddBits;
    }
}
```



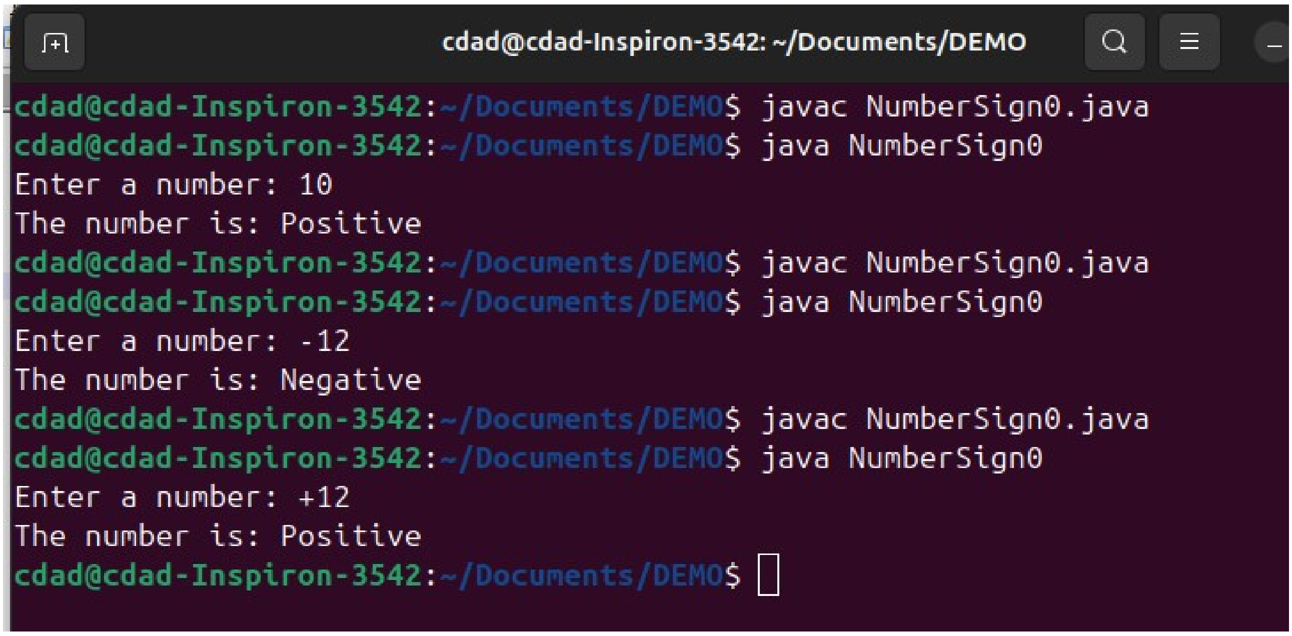
```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac BitSwap.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java BitSwap
Original number: 29 (binary: 11101)
Swapped number: 46 (binary: 101110)
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

4. Ternary Operator Challenges.

Q16:Write a program that determines whether a given number is positive, negative, or zero using only the ternary operator.

Ans.

```
import java.util.Scanner;

public class NumberSign0{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        double number = scanner.nextDouble();

        // Use the ternary operator to determine the sign of the number
                String result = (number > 0) ? "Positive" : (number < 0) ? "Negative" : "Zero";

        System.out.println("The number is: " + result);

        scanner.close();
    }
}
```


cdad@cdad-Inspiron-3542: ~/Documents/DEMO

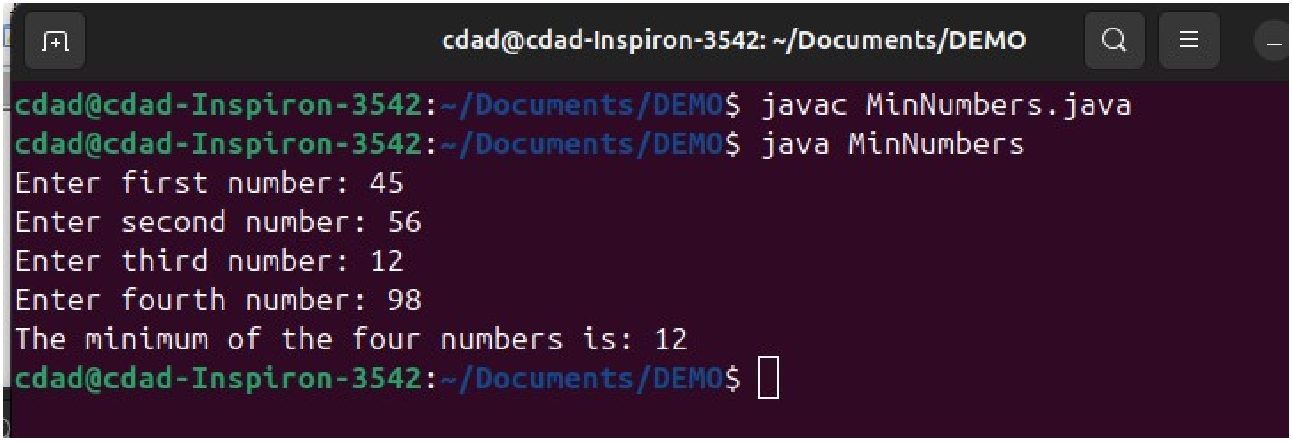
```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac NumberSign0.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java NumberSign0
Enter a number: 10
The number is: Positive
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac NumberSign0.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java NumberSign0
Enter a number: -12
The number is: Negative
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac NumberSign0.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java NumberSign0
Enter a number: +12
The number is: Positive
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

Q17:Implement a Java program that finds the minimum of four numbers using nested ternary operators.

Ans.

```
import java.util.Scanner;
```

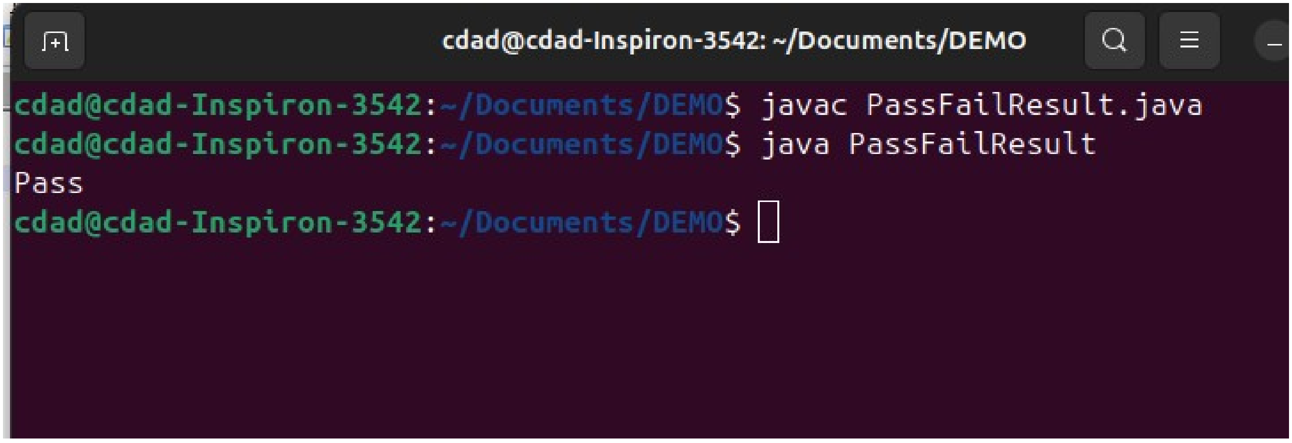
```
public class MinNumbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first number: ");
        int num1 = scanner.nextInt();

        System.out.print("Enter second number: ");
        int num2 = scanner.nextInt();

        System.out.print("Enter third number: ");
        int num3 = scanner.nextInt();

        System.out.print("Enter fourth number: ");
        int num4 = scanner.nextInt();

        // Finding minimum using nested ternary operator
        int min = (num1 < num2) ?
                        (num1 < num3 ? (num1 < num4 ? num1 : num4) : (num3 <
num4 ? num3 : num4)) :
                                              (num2 < num3 ? (num2 < num4 ? num2 :
num4) : (num3 < num4 ? num3 : num4));

        System.out.println("The minimum of the four numbers is: " + min);

        scanner.close();
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac MinNumbers.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java MinNumbers
Enter first number: 45
Enter second number: 56
Enter third number: 12
Enter fourth number: 98
The minimum of the four numbers is: 12
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

Q18:Given a student’s percentage, print“Pass”if the percentage is 40 or above; otherwise, print“Fail”, using only the ternary operator.
Ans.

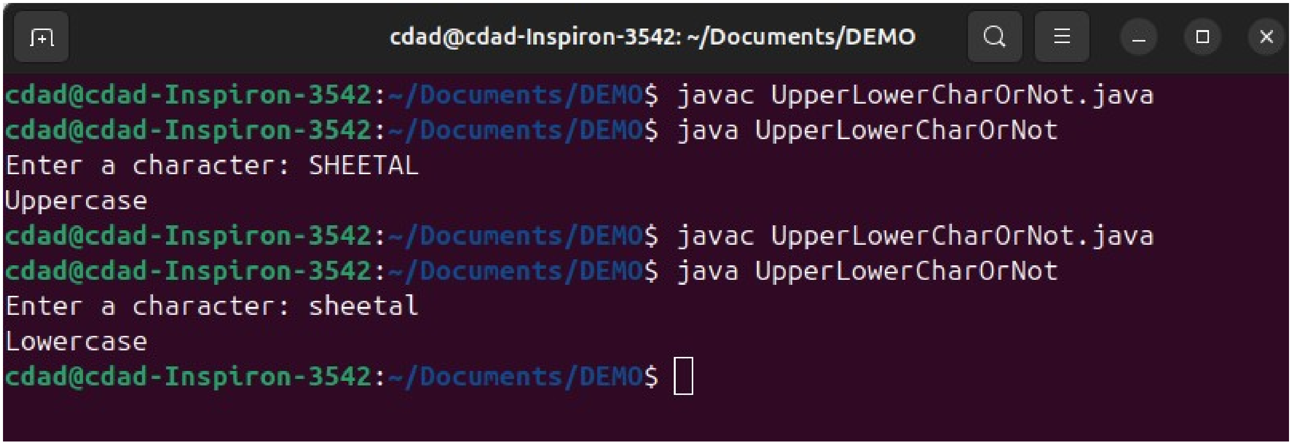
```
public class PassFailResult {
    public static void main(String[] args) {
        int percentage = 65;

        // Using the ternary operator
        String result = (percentage >= 40) ? "Pass" : "Fail";

        System.out.println(result);
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac PassFailResult.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java PassFailResult
Pass
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

Q19:Write a Java program that checks whether a character is uppercase, lowercase, or not a letter using only the ternary operator.
Ans.

```
import java.util.Scanner;

public class UpperLowerCharOrNot {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a character: ");
        char ch = scanner.next().charAt(0);
        String result = Character.isUpperCase(ch) ? "Uppercase" :
                        Character.isLowerCase(ch) ? "Lowercase" :
                        "Not a letter";

        System.out.println(result);

        scanner.close();
    }
}
```

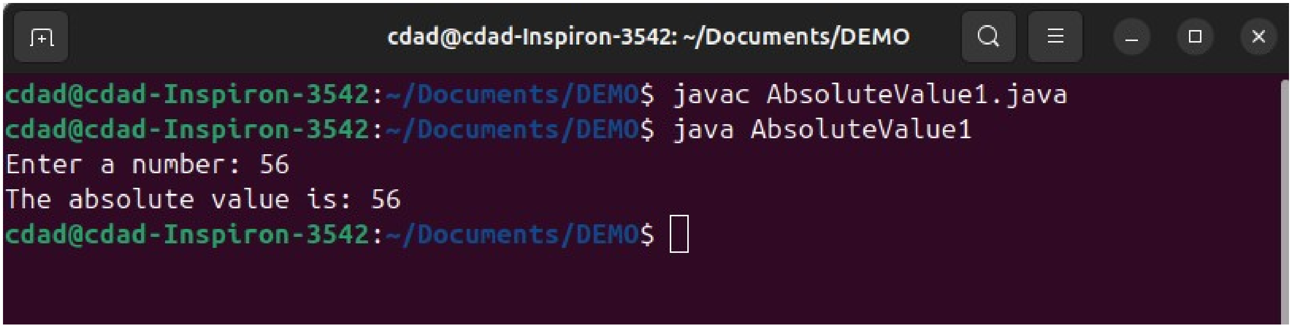


cdad@cdad-Inspiron-3542: ~/Documents/DEMO

```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac UpperLowerCharOrNot.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java UpperLowerCharOrNot
Enter a character: SHEETAL
Uppercase
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac UpperLowerCharOrNot.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java UpperLowerCharOrNot
Enter a character: sheetal
Lowercase
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

Q20:Implement a Java program that returns the absolute value of a given number using the ternary operator (without using Math.abs()).
Ans.

```
import java.util.Scanner;

public class AbsoluteValue1{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

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

        // Using the ternary operator to get the absolute value
        int absoluteValue = (number >= 0) ? number : -number;

        System.out.println("The absolute value is: " + absoluteValue);

        scanner.close();
    }
}
```



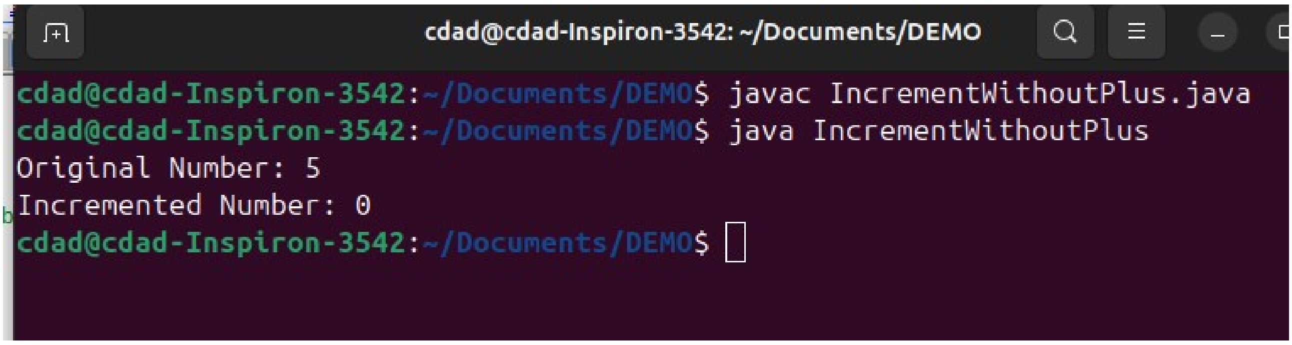
cdad@cdad-Inspiron-3542: ~/Documents/DEMO

```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac AbsoluteValue1.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java AbsoluteValue1
Enter a number: 56
The absolute value is: 56
cdad@cdad-Inspiron-3542:~/Documents/DEMO$
```

5. Miscellaneous Operator Questions

Q21:Write a program that increments a number without using + or ++operators.
Hint: Use bitwise- (~x).
Ans.

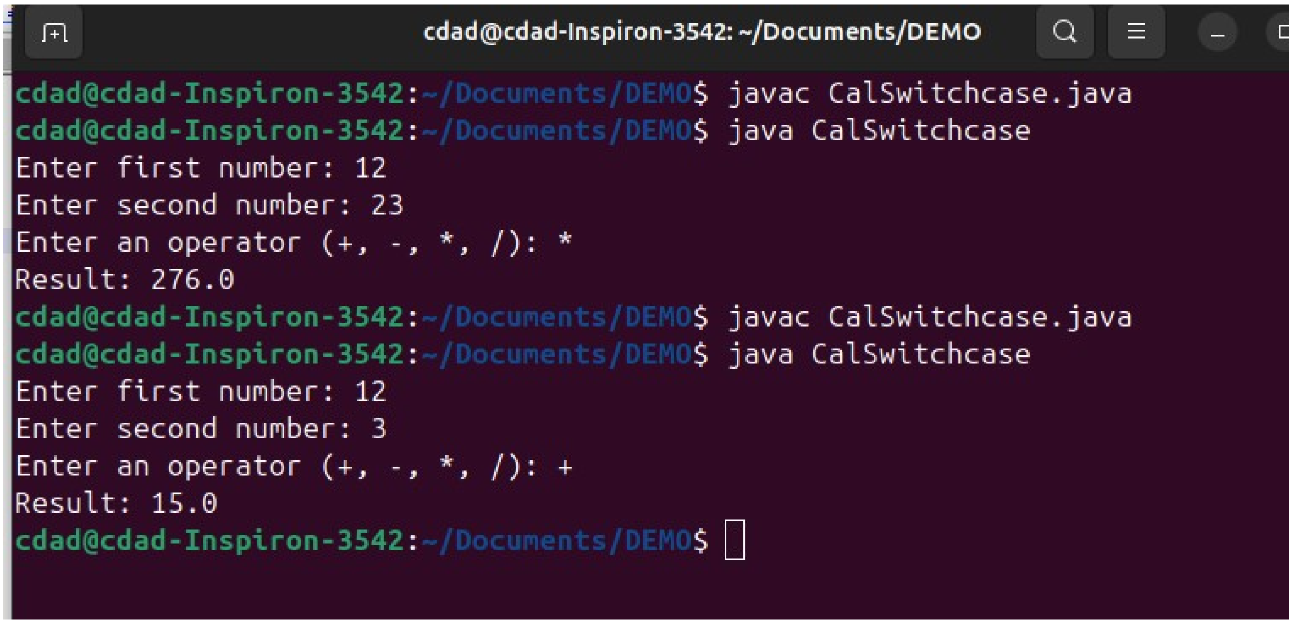
```
public class IncrementWithoutPlus {
    public static int increment(int number) {
        int temp;

        while ((temp = number & -number) != 0) {
            number = number ^ temp; // Number without the least significant
set bit
            temp = (temp << 1);      // Shift the least significant set bit
            number = number | temp;  // Add the shifted bit to the number
        }

        return number;
    }

    public static void main(String[] args) {
        int number = 5; // Example number to increment
        int incrementedNumber = increment(number);
        System.out.println("Original Number: " + number);
        System.out.println("Incremented Number: " + incrementedNumber);
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac IncrementWithoutPlus.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java IncrementWithoutPlus
Original Number: 5
Incremented Number: 0
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

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

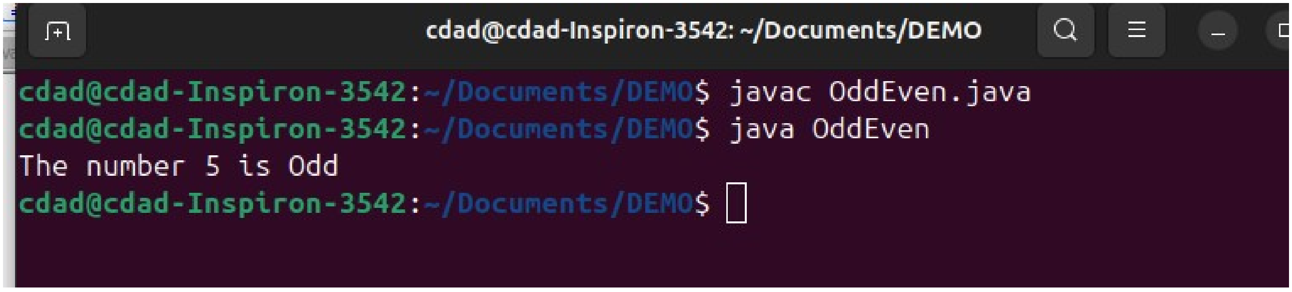
```
import java.util.Scanner;

public class CalSwitchcase{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first number: ");
        double firstNumber = scanner.nextDouble();

        System.out.print("Enter second number: ");
        double secondNumber = scanner.nextDouble();

        System.out.print("Enter an operator (+, -, *, /): ");
        char operator = scanner.next().charAt(0);

        double result;

        // Switch-case statement to perform the operation
                 switch (operator) {
            case '+':
                result = firstNumber + secondNumber;
                System.out.println("Result: " + result);
                break;
            case '-':
                result = firstNumber - secondNumber;
                System.out.println("Result: " + result);
                break;

            case '*':
                result = firstNumber * secondNumber;
                System.out.println("Result: " + result);
break;

            case '/':
                // Handle division by zero
                if (secondNumber != 0) {
                    result = firstNumber / secondNumber;
                    System.out.println("Result: " + result);
                } else {
                                    System.out.println("Error: Division by zero is not
allowed.");
                }
                break;

            default:
                System.out.println("Error: Invalid operator. Please use one of (+,
-, *, /).");
                break;
        }

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



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac CalSwitchcase.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java CalSwitchcase
Enter first number: 12
Enter second number: 23
Enter an operator (+, -, *, /): *
Result: 276.0
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac CalSwitchcase.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java CalSwitchcase
Enter first number: 12
Enter second number: 3
Enter an operator (+, -, *, /): +
Result: 15.0
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

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

```
public class OddEven {
    public static void main(String[] args) {
        int number = 5;
        String[] result = {"Even", "Odd"}; // Array to hold the results

        int index = number & 1; // 0 for even, 1 for odd

        System.out.println("The number " + number + " is " + result[index]);
    }
}
```

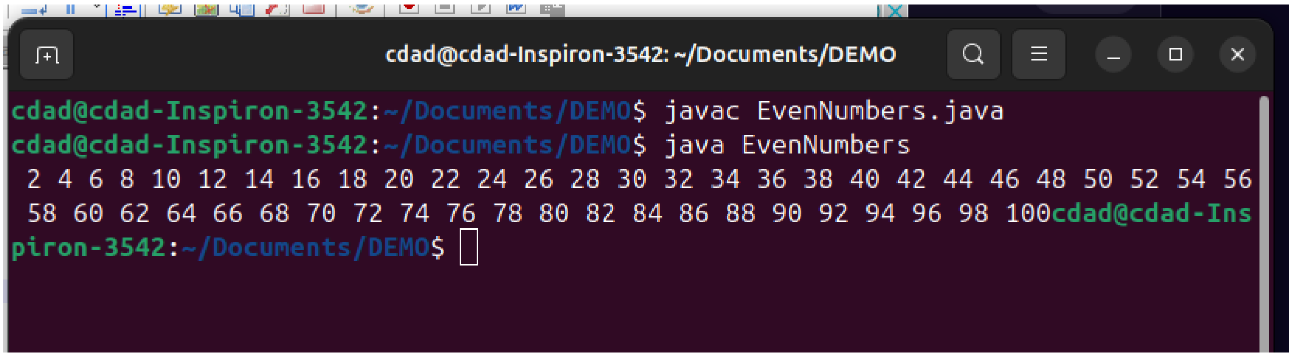


```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac OddEven.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java OddEven
The number 5 is Odd
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```

Q24:Write a program that prints all even numbers from 1 to 100using only bitwise AND(&)and for loop.

Ans.

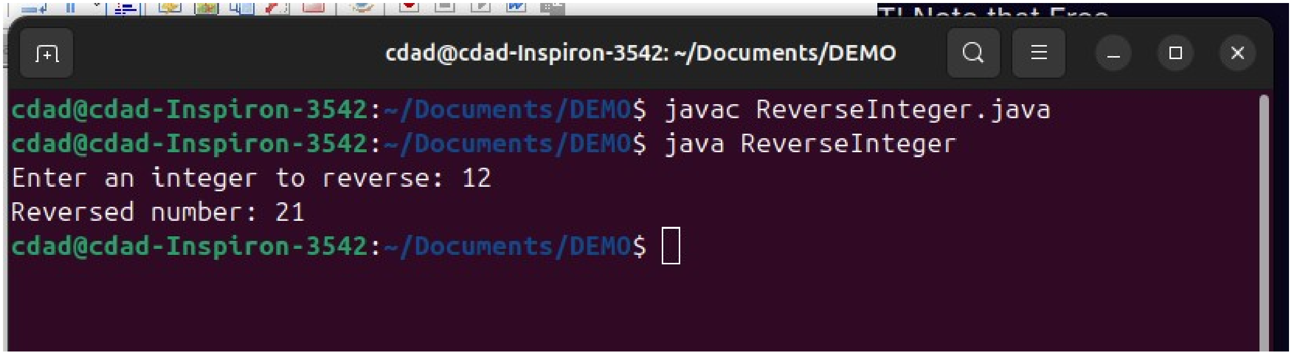
```
public class EvenNumbers {
    public static void main(String[] args) {

        for (int i = 1; i <= 100; i++) {
            // Check if the number is even using bitwise AND
            if ((i & 1) == 0) {
                System.out.print(" "+i);
            }
        }
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac EvenNumbers.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java EvenNumbers
 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 100cdad@cdad-Ins
piron-3542:~/Documents/DEMO$ 
```

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;}
Ans.

```
import java.util.Scanner;

public class ReverseInteger {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

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

        int reversed = reverseInteger(number);

        System.out.println("Reversed number: " + reversed);
    }
    public static int reverseInteger(int n) {
         int rev = 0;

        while (n != 0) {
            // Get the last digit and add it to rev
            rev = rev * 10 + n % 10;

            // Remove the last digit from n
            n /= 10;
        }

        return rev;
    }
}
```



```
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ javac ReverseInteger.java
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ java ReverseInteger
Enter an integer to reverse: 12
Reversed number: 21
cdad@cdad-Inspiron-3542:~/Documents/DEMO$ 
```