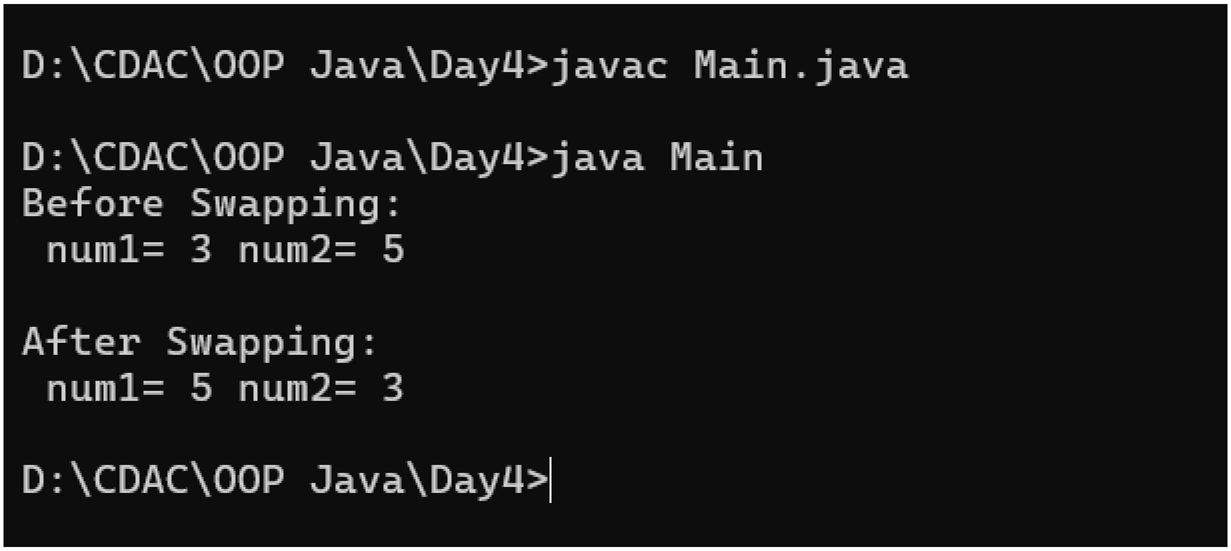
Name-Aarya Sanjay Dange

Assignment 02 (OOPJ)

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.

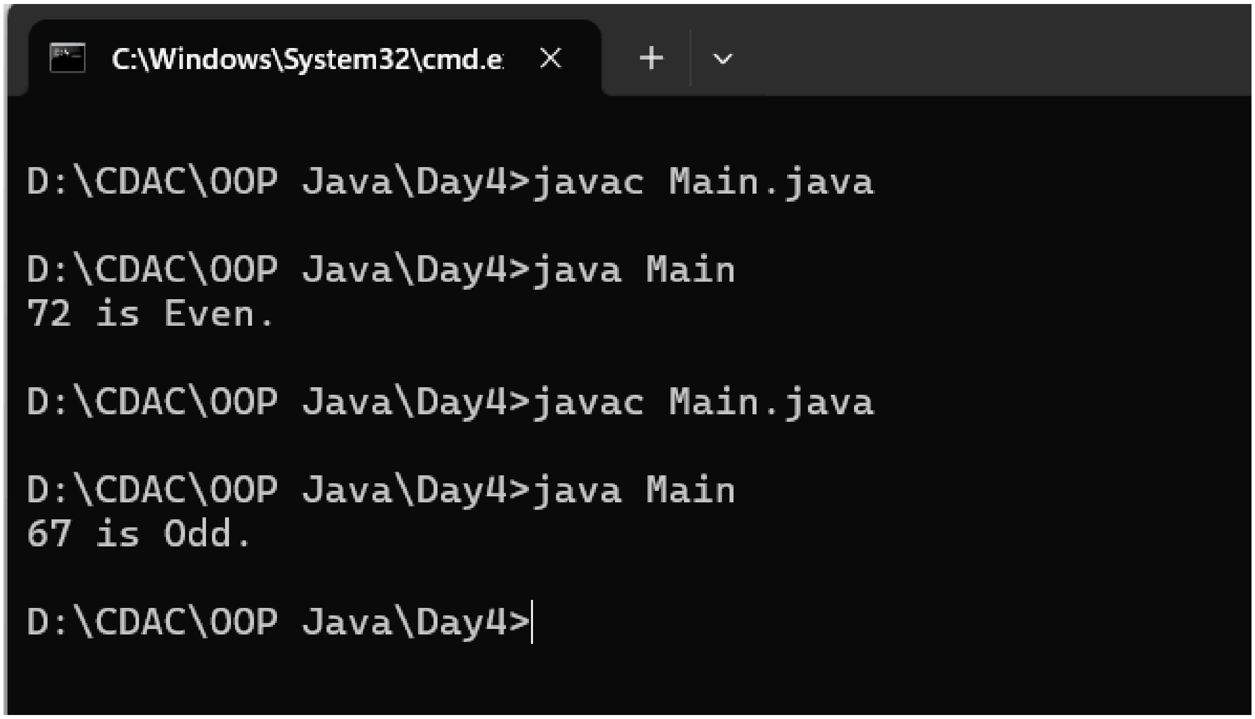
```
class Main{
    public static void main(String args[]){

            int num1=3, num2=5;
            System.out.println("Before Swapping:\n
num1= "+num1+" num2= "+num2);

            num1=num1^num2;
            num2=num1^num2;
            num1=num1^num2;
            System.out.println("\nAfter Swapping:\n
num1= "+num1+" num2= "+num2);

      }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Before Swapping:
 num1= 3 num2= 5

After Swapping:
 num1= 5 num2= 3

D:\CDAC\OOP Java\Day4>
```

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

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

            int num1=67;
            if((num1 & 1) %2 ==0){
                System.out.println(num1+" is Even.");
            }
            else{
                System.out.println(num1+" is Odd.");
            }
      }
}
```

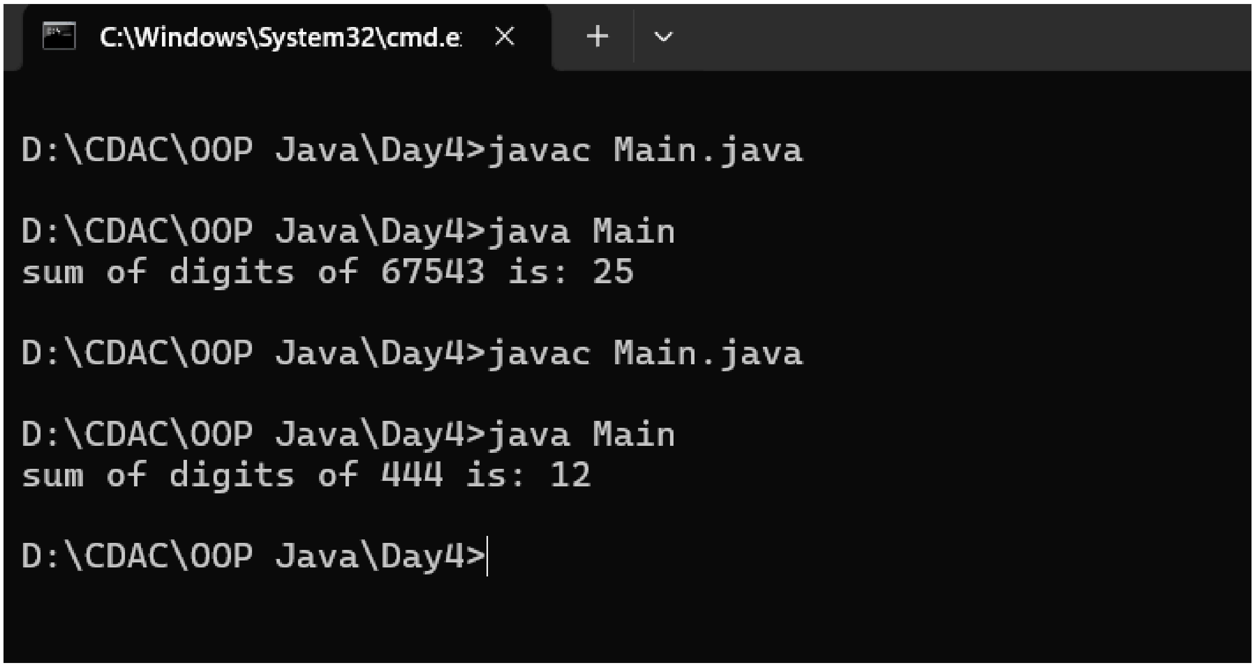


```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
72 is Even.

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
67 is Odd.

D:\CDAC\OOP Java\Day4>
```

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

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

            int num1=444;
            int num2=num1;
            int sum=0;
            int rem;
            while(num1>0){
```

```
            rem=num1%10;
            sum=sum+rem;
            num1=num1/10;
        }
        System.out.println("sum of digits of
"+num2+" is: "+sum);
    }
}
```

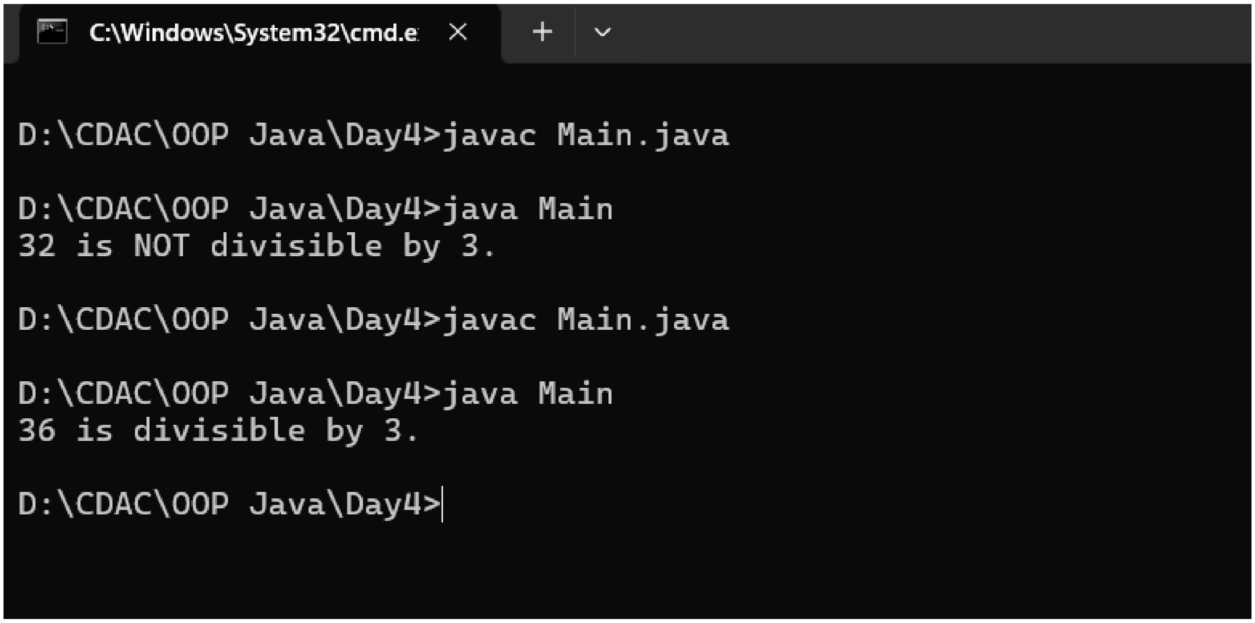


```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
sum of digits of 67543 is: 25

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
sum of digits of 444 is: 12

D:\CDAC\OOP Java\Day4>
```

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 .

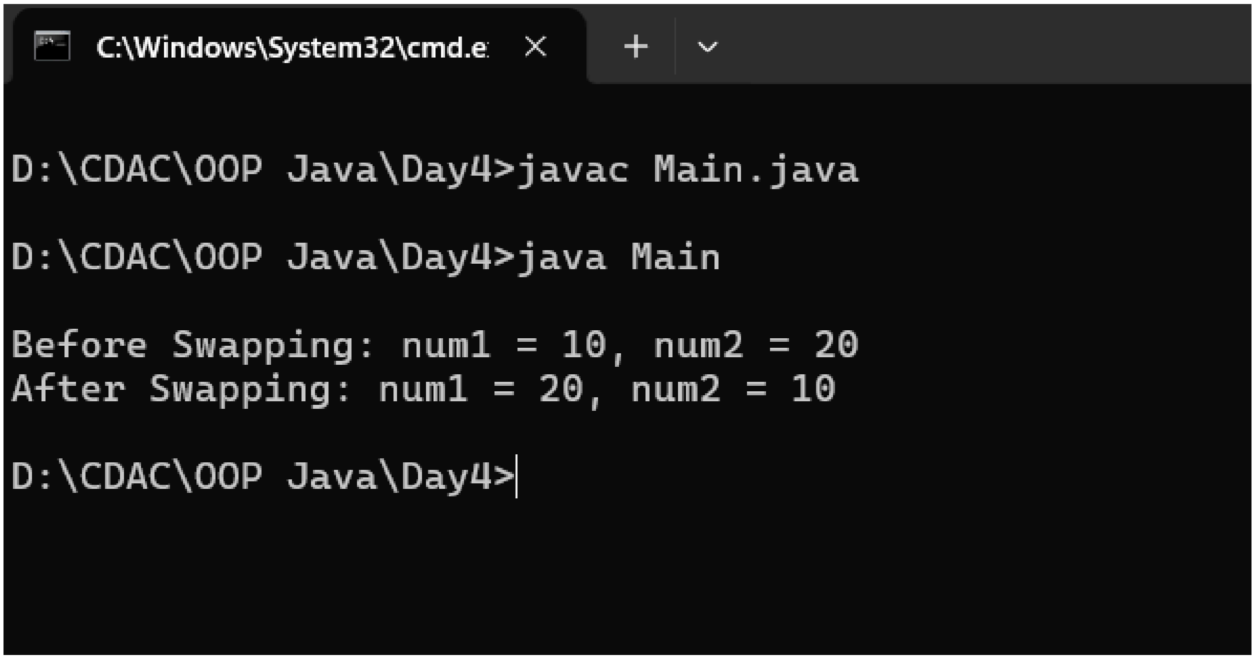
```
class Main{

    static boolean isDivisibleBy3(int n) {
        int oddSum = 0, evenSum = 0;
        boolean isOdd = true;

        while (n != 0) {
            if (isOdd) {
                oddSum += (n & 1);
            } else {
                evenSum += (n & 1);
            }
            isOdd = !isOdd;
            n >>= 1;
        }

        int difference = Math.abs(oddSum - evenSum);
        return (difference == 0 || difference % 3 == 0);
    }

    public static void main(String[] args) {
        int num =36;
        if (isDivisibleBy3(num)) {
            System.out.println(num + " is divisible by 3.");
        } else {
            System.out.println(num + " is NOT divisible by 3.");
        }
    }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
32 is NOT divisible by 3.

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
36 is divisible by 3.

D:\CDAC\OOP Java\Day4>
```

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

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

        int num1 =10;
        int num2 = 20;
        System.out.println("\nBefore Swapping: num1 = " +
num1 + ", num2 = " + num2);
        num1 += num2;
        num2 = num1 - num2;
        num1 -= num2;
        System.out.println("After Swapping: num1 = " + num1
+ ", num2 = " + num2);
    }
}
```

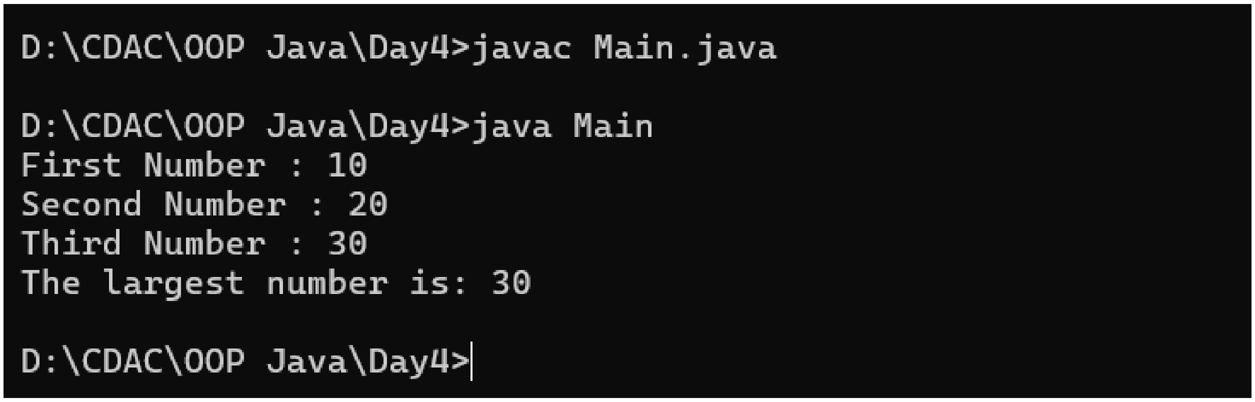


```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main

Before Swapping: num1 = 10, num2 = 20
After Swapping: num1 = 20, num2 = 10

D:\CDAC\OOP Java\Day4>
```

2. Relational & Logical Operators :

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

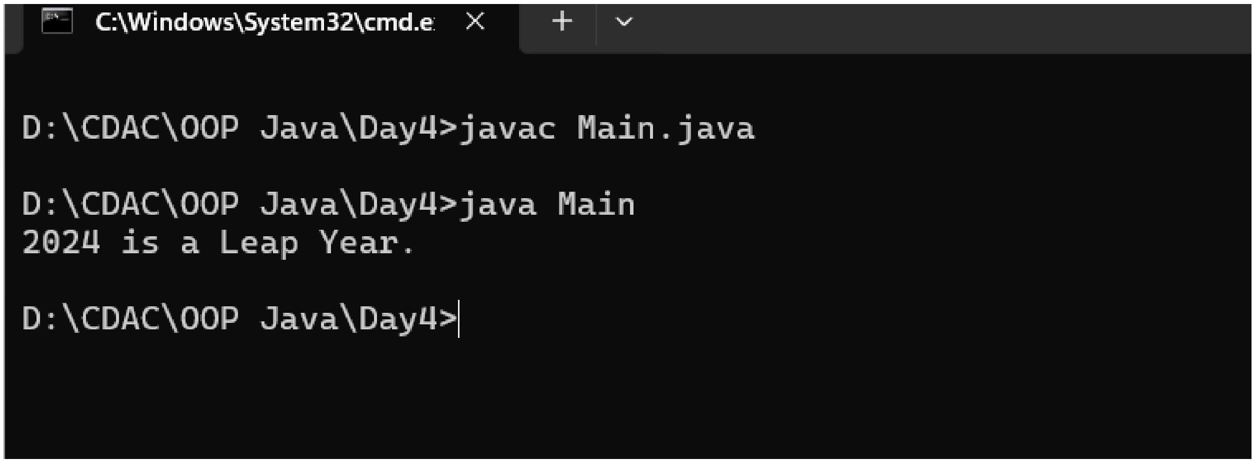
```
class Main{
    public static void main(String[] args) {

        int num1 =10;
        int num2 = 20;
        int num3 = 30;

        System.out.println("First Number : "+num1);
        System.out.println("Second Number : "+num2);
        System.out.println("Third Number : "+num3);

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



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
First Number : 10
Second Number : 20
Third Number : 30
The largest number is: 30

D:\CDAC\OOP Java\Day4>
```

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

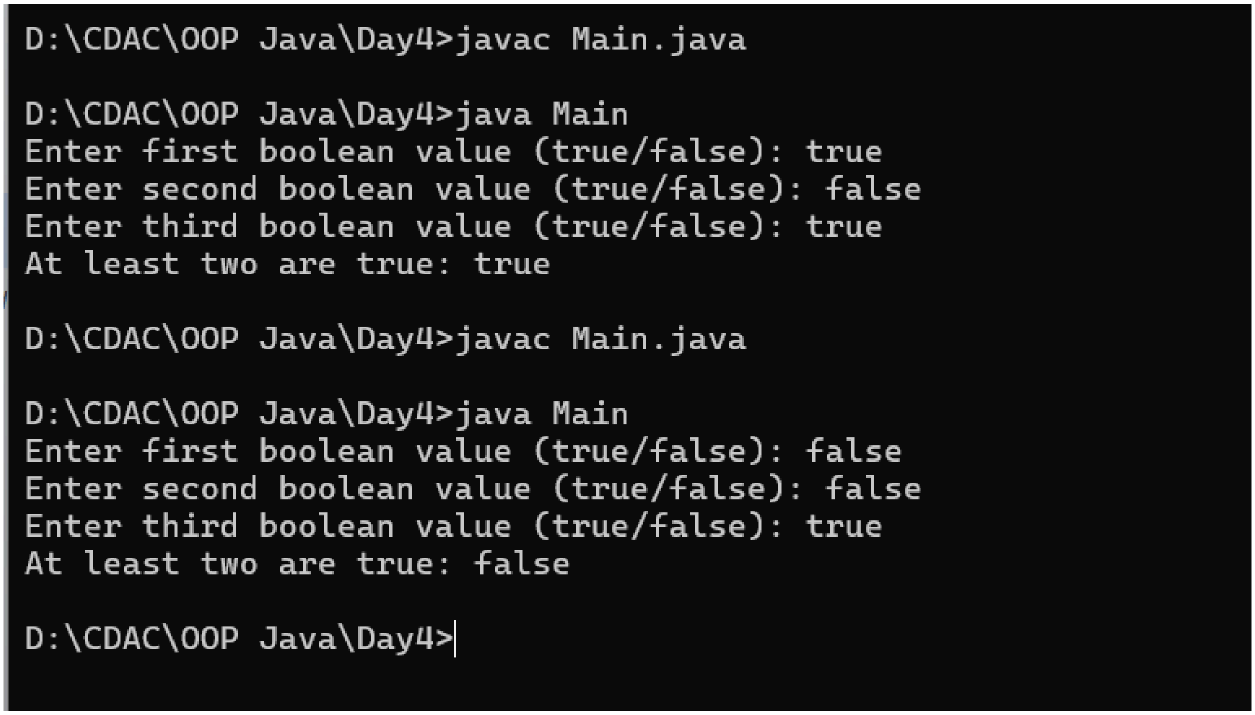
```
class Main{
   public static void main(String[] args) {

       int year =2024;

       if ((year % 400 == 0) || ((year % 4 == 0) && (year % 100 != 0))) {
           System.out.println(year + " is a Leap Year.");
       } else {
           System.out.println(year + " is NOT a Leap Year.");
       }
   }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
2024 is a Leap Year.

D:\CDAC\OOP Java\Day4>
```

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

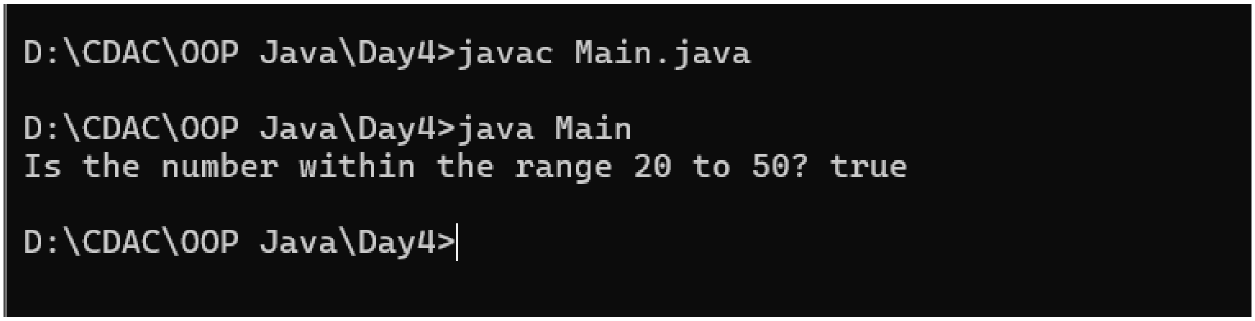
```
import java.util.Scanner;

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

        Scanner sc = new Scanner(System.in);

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

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

        boolean result = (a && b) || (a && c) || (b && c);

        System.out.println("At least two are true: " + result);
    }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Enter first boolean value (true/false): true
Enter second boolean value (true/false): false
Enter third boolean value (true/false): true
At least two are true: true

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Enter first boolean value (true/false): false
Enter second boolean value (true/false): false
Enter third boolean value (true/false): true
At least two are true: false

D:\CDAC\OOP Java\Day4>
```

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 .

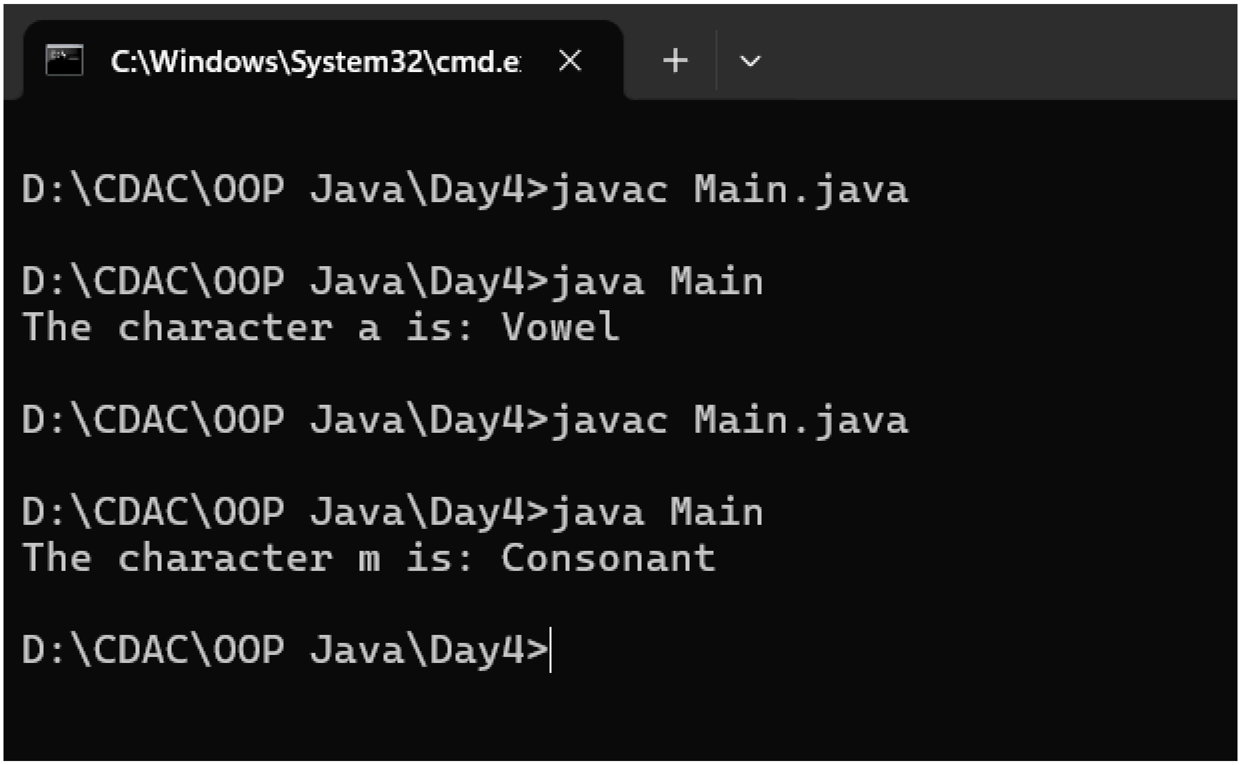
```
class Main{
   public static void main(String[] args) {

       int num =25;
       System.out.println("Is the number within the range 20 to 50? " + (num >= 20 && num <= 50));
   }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Is the number within the range 20 to 50? true

D:\CDAC\OOP Java\Day4>
```

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

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

       char ch ='m';
       String result = (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'|| ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' ||
ch == 'U')? "Vowel": ((ch >= 'a' && ch <= 'z') ? "Consonant"
: "Invalid input");
        System.out.println("The character "+ch+" is: " +
result);
    }
}
```

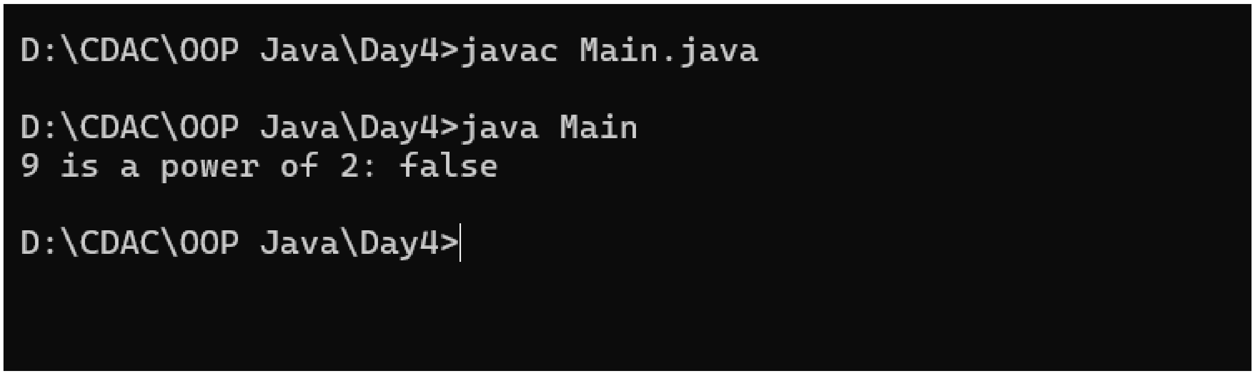


```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
The character a is: Vowel

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
The character m is: Consonant

D:\CDAC\OOP Java\Day4>
```

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.

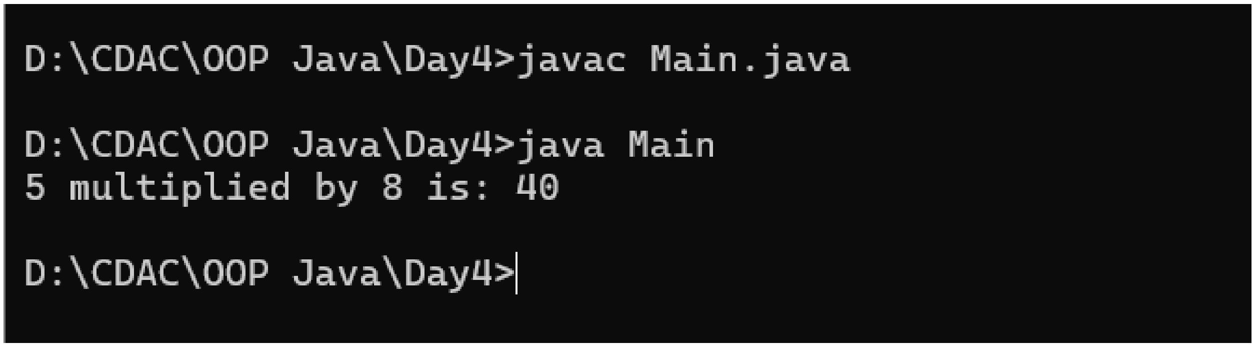
```
class Main{
    public static void main(String[] args) {
          int n = 9;

        boolean isPowerOfTwo = (n > 0) && ((n & (n - 1)) ==
0);
        System.out.println(n + " is a power of 2: " +
isPowerOfTwo);
    }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
9 is a power of 2: false

D:\CDAC\OOP Java\Day4>
```

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

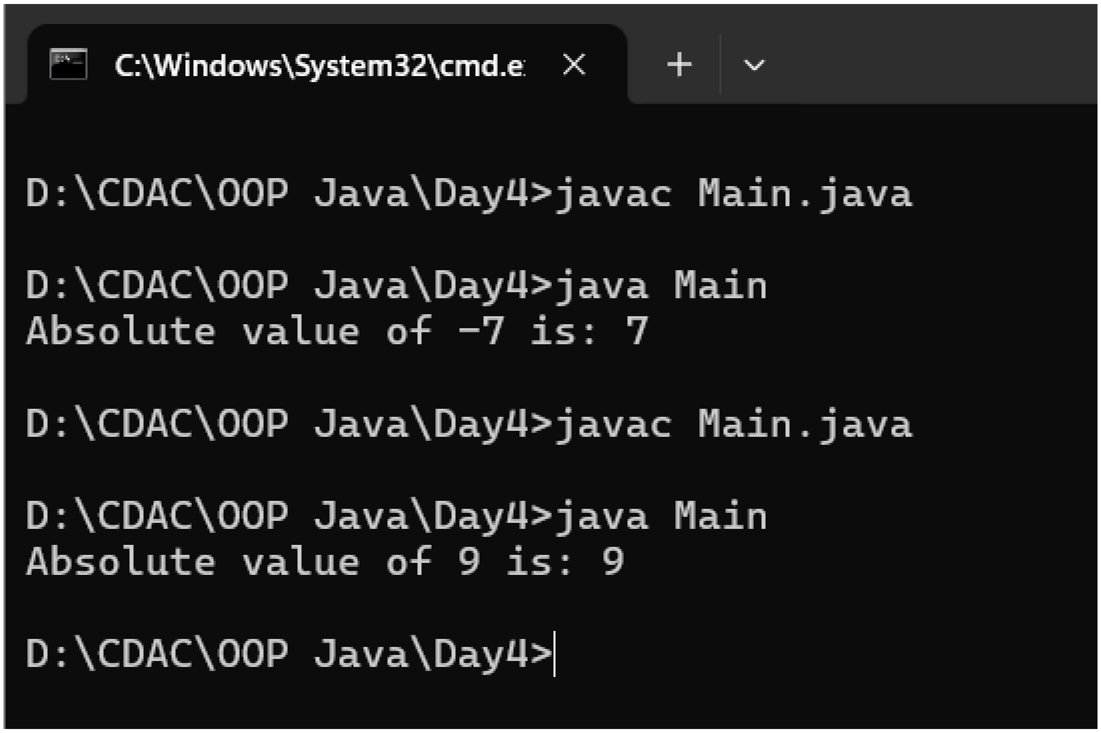
```
class Main{
   public static void main(String[] args) {
      int num =5;
      int result = num << 3;

      System.out.println(num + " multiplied by 8 is: " +
result);
   }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
5 multiplied by 8 is: 40

D:\CDAC\OOP Java\Day4>
```

Q13: Implement a Java program to find the absolute value of an integer using bitwise operators. Hint : mask = num >> 31; abs = (num + mask) ^ mask;

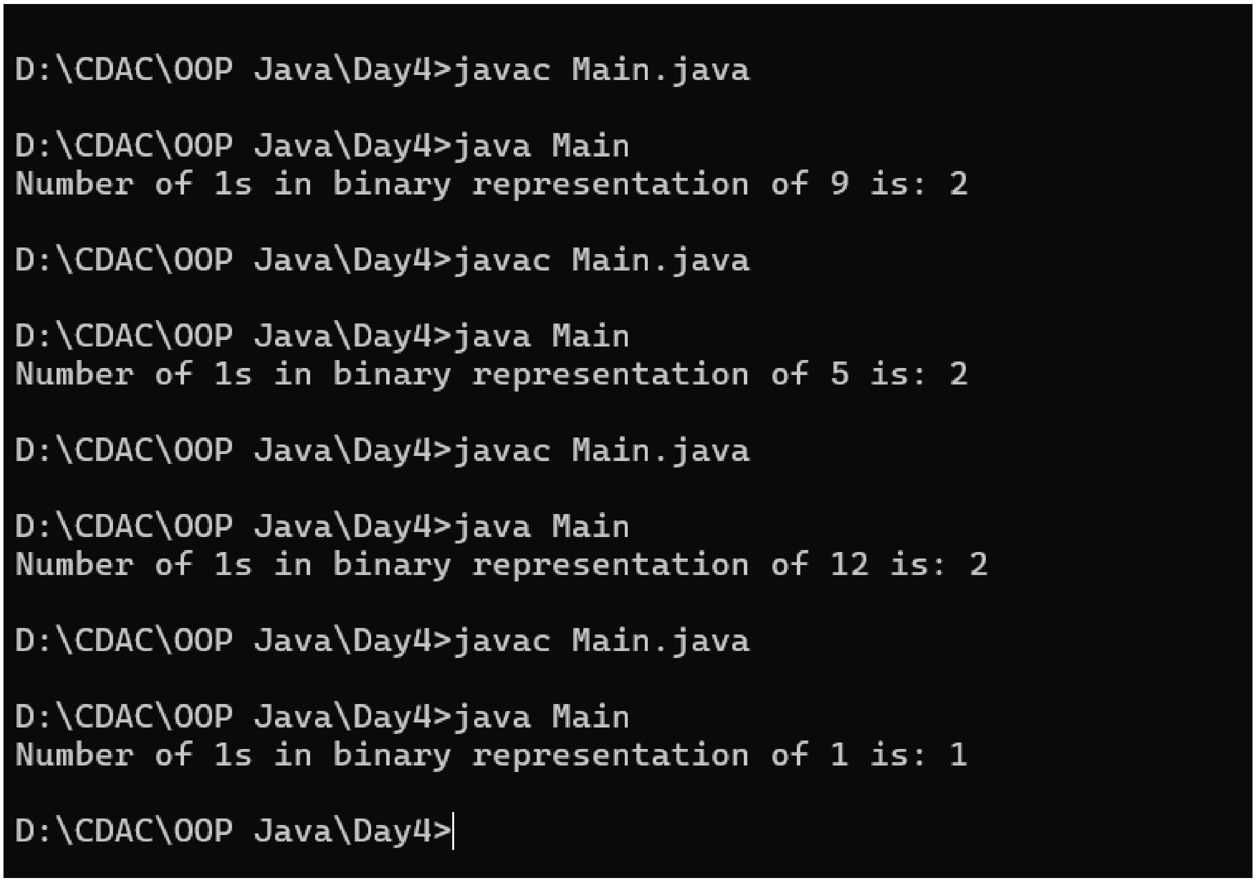
```
class Main{
    public static void main(String[] args) {
        int num =9;

        int mask = num >> 31;
        int abs = (num + mask) ^ mask;

        System.out.println("Absolute value of "+num+" is: "+
abs);
    }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Absolute value of -7 is: 7

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Absolute value of 9 is: 9

D:\CDAC\OOP Java\Day4>
```

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) .

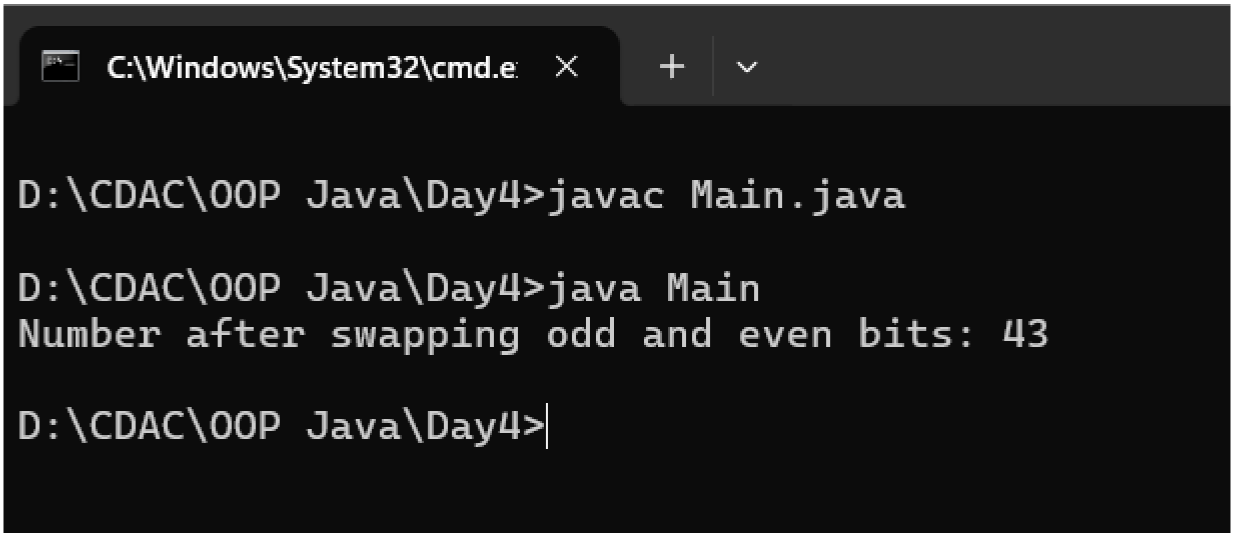
```
class Main{
    public static void main(String[] args) {
        int num =1;
       int num2=num;
        int count = 0;
        while (num > 0) {
            num = num & (num - 1);
            count++;
        }

        System.out.println("Number of 1s in binary
representation of "+num2+" is: "+ count);    }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Number of 1s in binary representation of 9 is: 2

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Number of 1s in binary representation of 5 is: 2

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Number of 1s in binary representation of 12 is: 2

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Number of 1s in binary representation of 1 is: 1

D:\CDAC\OOP Java\Day4>
```

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

```
class Main{
    public static void main(String[] args) {
        int n =23;
       int evenBits = (n & 0xAAAAAAAA) >> 1;
        int oddBits = (n & 0x55555555) << 1;
        int swapped = evenBits | oddBits;

        System.out.println("Number after swapping odd and even bits: " + swapped);
}
}
```



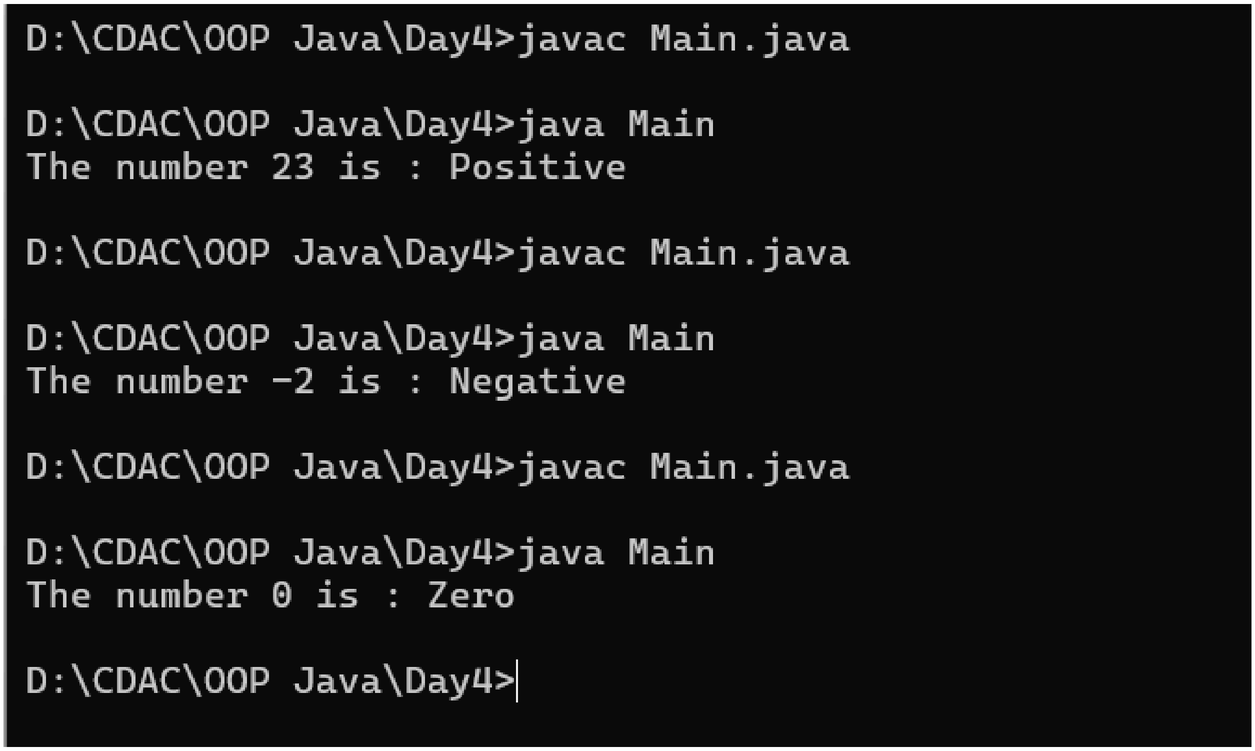
```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Number after swapping odd and even bits: 43

D:\CDAC\OOP Java\Day4>
```

4. Ternary Operator Challenges

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

```
class Main{
    public static void main(String[] args) {
        int num =0;
       String result = (num > 0) ? "Positive" : (num < 0) ?
"Negative" : "Zero";

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

}

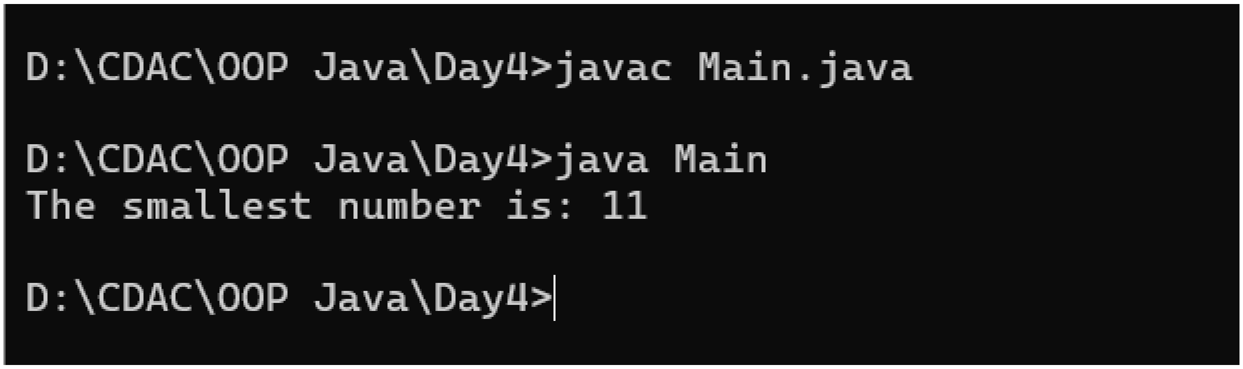


```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
The number 23 is : Positive

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
The number -2 is : Negative

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
The number 0 is : Zero

D:\CDAC\OOP Java\Day4>
```

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

```
class Main{
    public static void main(String[] args) {
       int a =11;
        int b = 14;
        int c = 22;
        int d = 33;

        int small = (a < b) ? ((a < c) ? ((a < d) ? a : d) : (c < d
? c : d))
```

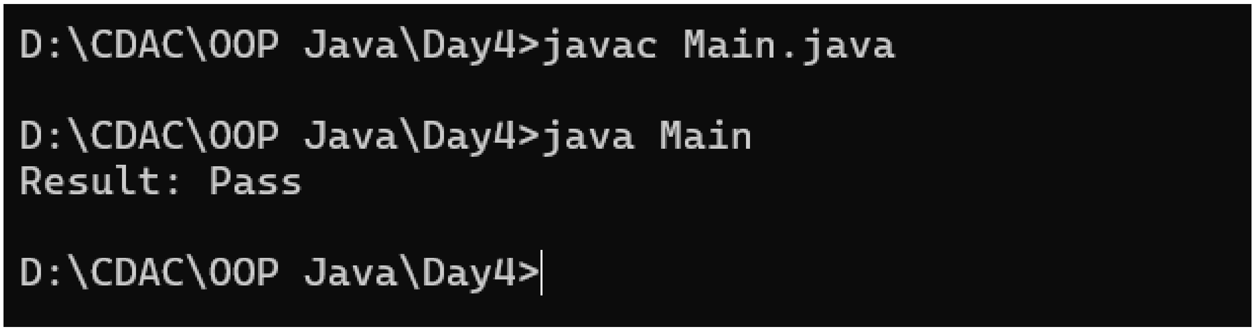
```
                    : ((b < c) ? ((b < d) ? b : d) : (c < d ? c :
d));

        System.out.println("The smallest number is: " +
small);
}
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
The smallest number is: 11

D:\CDAC\OOP Java\Day4>
```

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

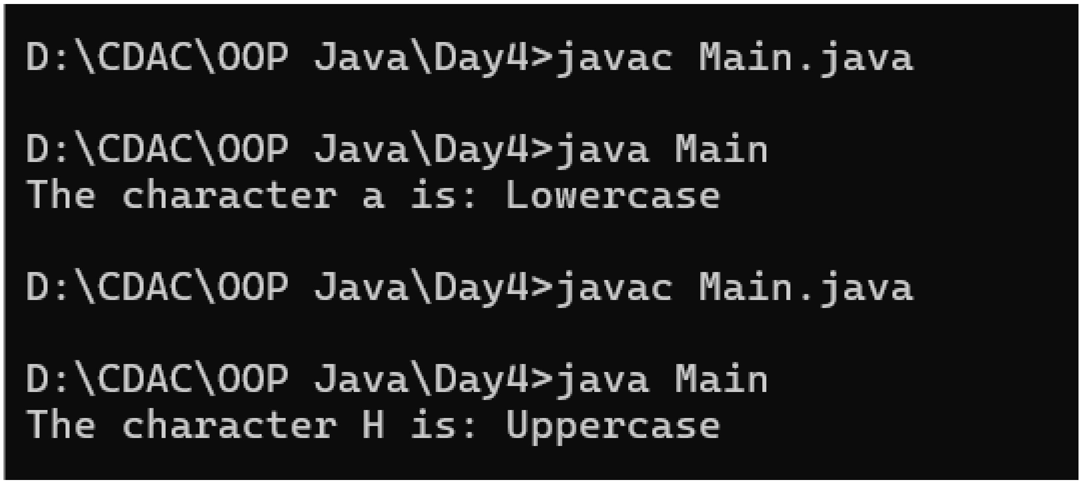
```
class Main{
   public static void main(String[] args) {
      int percentage = 61;

       String result = (percentage >= 40) ? "Pass" : "Fail";
       System.out.println("Result: " + result);
      }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Result: Pass

D:\CDAC\OOP Java\Day4>
```

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

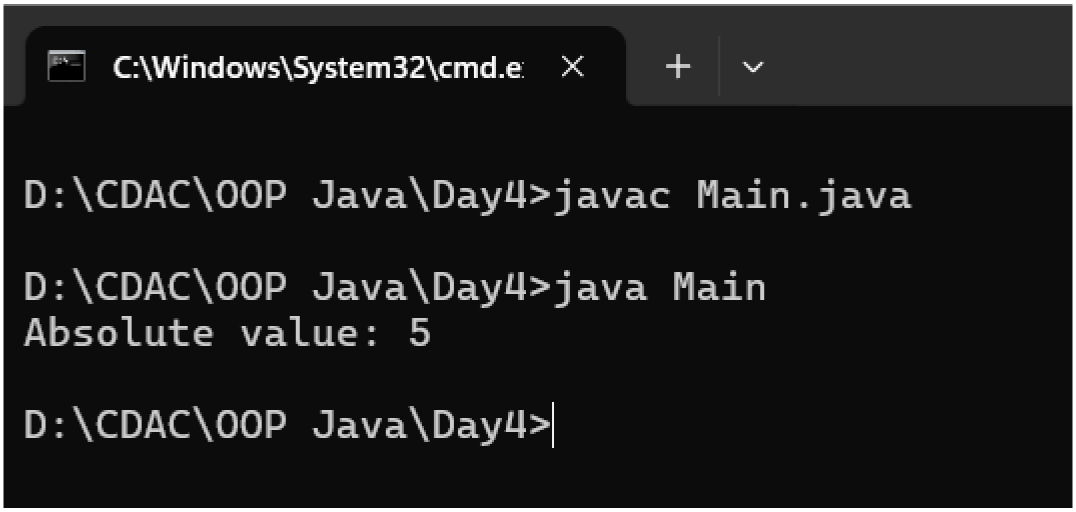
```
class Main{
   public static void main(String[] args) {
      char ch = '56';

       String result = (ch >= 'A' && ch <= 'Z') ? "Uppercase"
: (ch >= 'a' && ch <= 'z') ? "Lowercase": "Not a letter";

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



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
The character a is: Lowercase

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
The character H is: Uppercase
```

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

```
class Main{
    public static void main(String[] args) {
       int num=-5;
       int absValue = (num < 0) ? -num : num;

        System.out.println("Absolute value: " + absValue);
       }
}
```

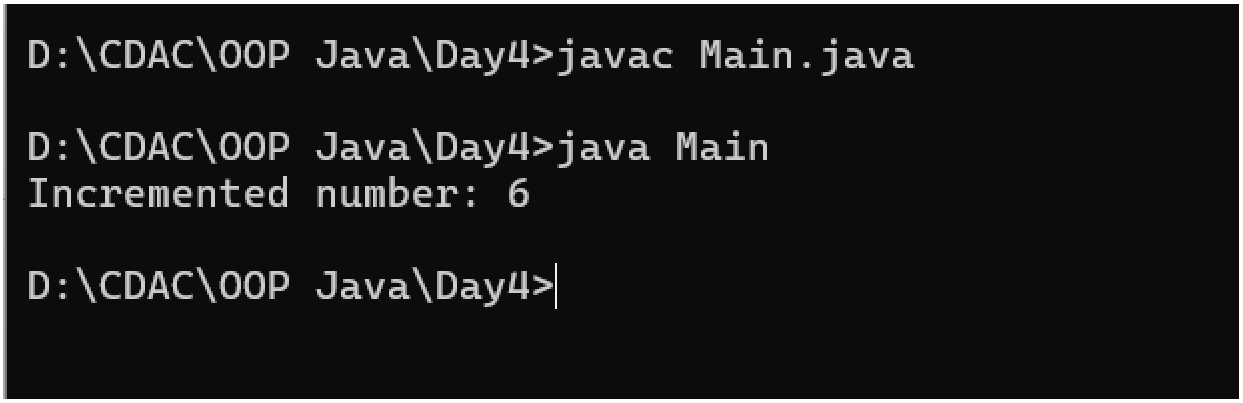


```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Absolute value: 5

D:\CDAC\OOP Java\Day4>
```

5. Miscellaneous Operator Questions

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

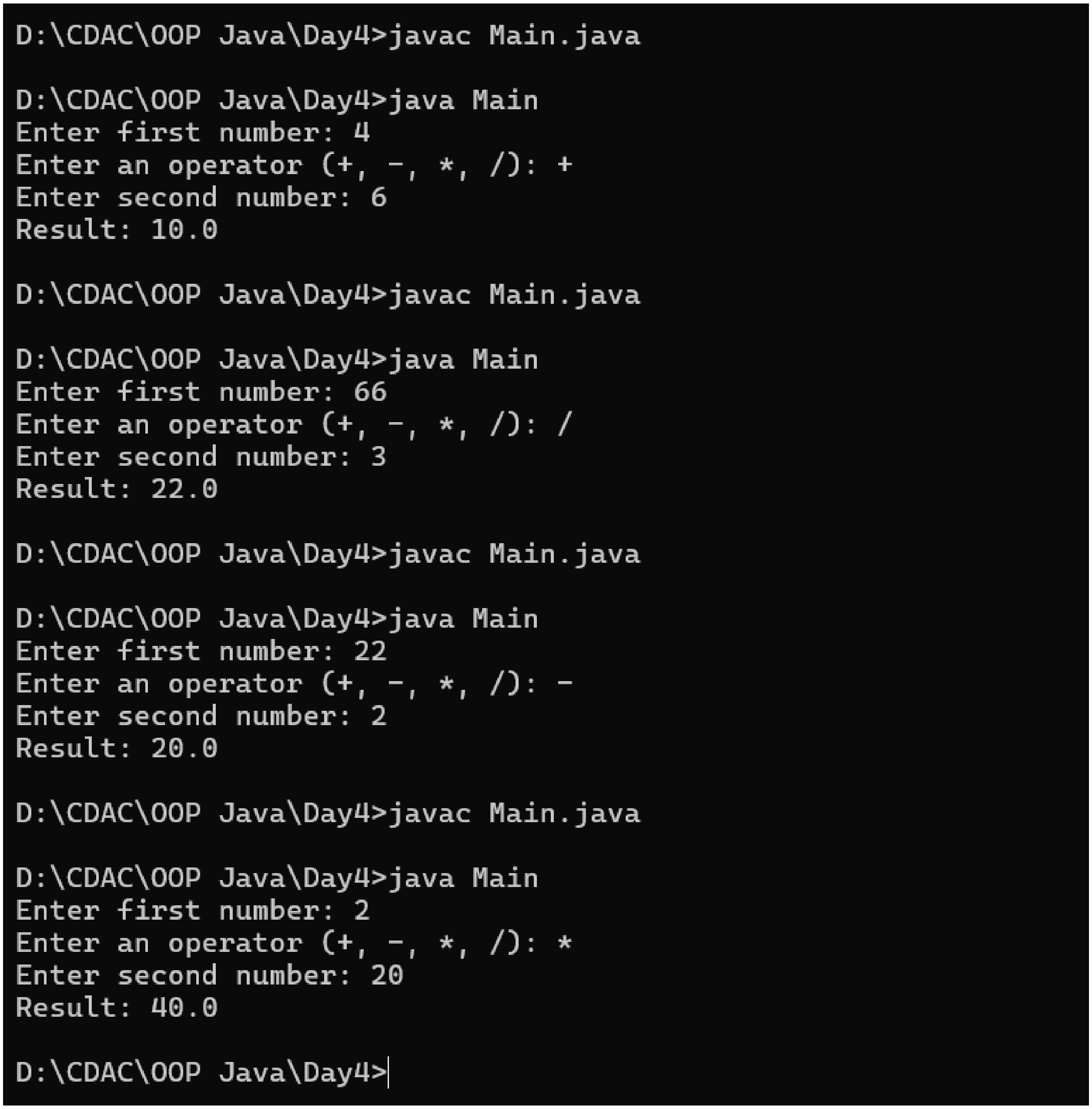
```
class Main{
    public static void main(String[] args) {
       int num=5;
        int incrementedNum = -(~num);
        System.out.println("Incremented number: " +
incrementedNum);
       }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Incremented number: 6

D:\CDAC\OOP Java\Day4>
```

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

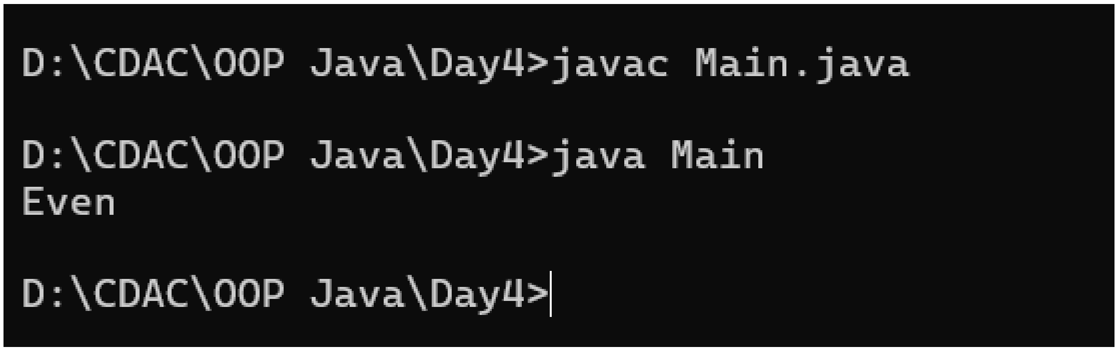
```
import java.util.Scanner;

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

        System.out.print("Enter first number: ");
        double num1 = scanner.nextDouble();

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

        System.out.print("Enter second number: ");
        double num2 = scanner.nextDouble();

        double result = 0;

        switch (operator) {
            case '+': result = num1 + num2; break;
            case '-': result = num1 - num2; break;
            case '*': result = num1 * num2; break;
            case '/': result = num2 != 0 ? num1 / num2 :
Double.NaN; break;
            default: System.out.println("Invalid operator!");
return;
        }

        System.out.println("Result: " + result);
}}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Enter first number: 4
Enter an operator (+, -, *, /): +
Enter second number: 6
Result: 10.0

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Enter first number: 66
Enter an operator (+, -, *, /): /
Enter second number: 3
Result: 22.0

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Enter first number: 22
Enter an operator (+, -, *, /): -
Enter second number: 2
Result: 20.0

D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Enter first number: 2
Enter an operator (+, -, *, /): *
Enter second number: 20
Result: 40.0

D:\CDAC\OOP Java\Day4>
```

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

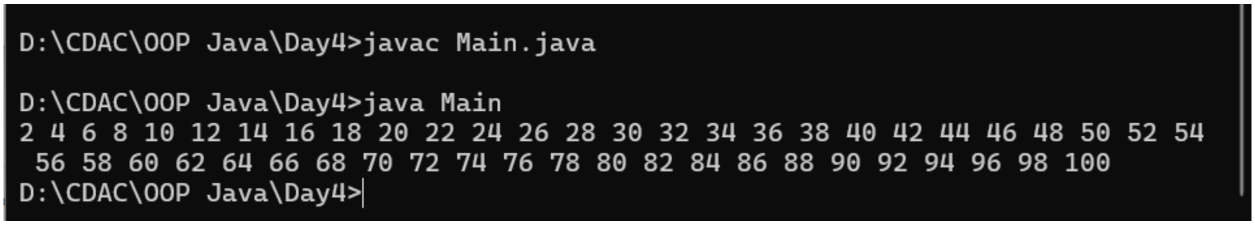
```
import java.util.Scanner;

class Main{
    public static void main(String[] args) {
       int num =12;

        System.out.println((num & 1) == 0 ? "Even" : "Odd");
}}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Even

D:\CDAC\OOP Java\Day4>
```

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

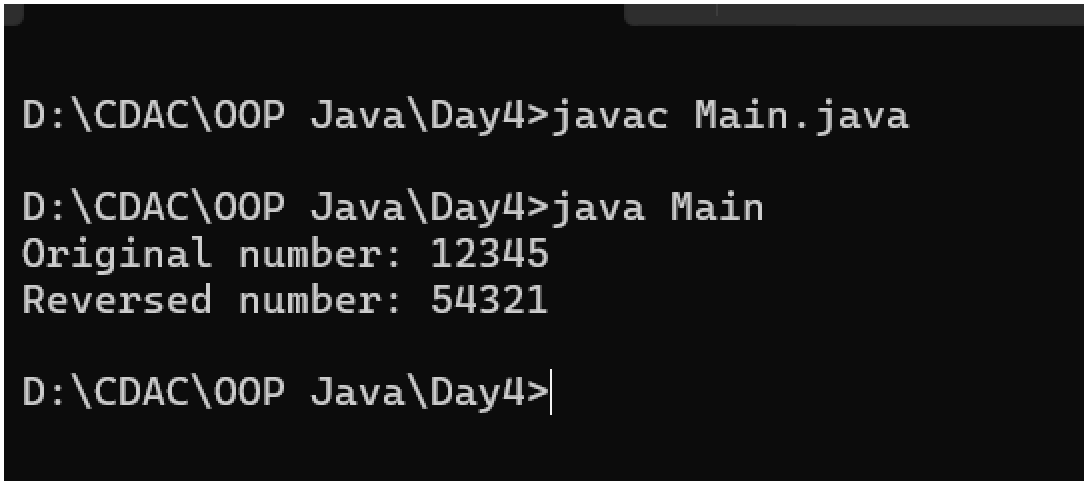
```
class Main{
    public static void main(String[] args) {
       for (int num = 1; num <= 100; num++) {
           System.out.print(((num & 1) == 0) ? num + " " : "");
        }
}}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
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
D:\CDAC\OOP Java\Day4>
```

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; }

```
import java.util.Scanner;

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

        int num =12345;
        int num2=num;
         int reversed = 0;

         while (num != 0) {
             reversed = reversed * 10 + num % 10;
             num =num/10;
         }

         System.out.println("Original number: " + num2);
         System.out.println("Reversed number: " + reversed);
    }
}
```



```
D:\CDAC\OOP Java\Day4>javac Main.java

D:\CDAC\OOP Java\Day4>java Main
Original number: 12345
Reversed number: 54321

D:\CDAC\OOP Java\Day4>
```