Assignment-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.

```
1.  Find the Largest and Smallest Element
        Given an array, find the smallest and largest elements in it.

*/

public class MinMaxFinder {
    public static void main(String[] args) {
        int[] arr = {10, 5, 20, 8, 25, 2, 15};  // Example array

        int min = arr[0];  // Initialize min with first element
        int max = arr[0];  // Initialize max with first element

        for (int i = 1; i < arr.length; i++) {
            if (arr[i] < min) {
                min = arr[i];  // Update min
            }
            if (arr[i] > max) {
                max = arr[i];  // Update max
            }
        }

        System.out.println("Smallest Element: " + min);
        System.out.println("Largest Element: " + max);
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac MinMaxFinder.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java MinMaxFinder
Smallest Element: 2
Largest Element: 25

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

import java.util.Scanner;

public class EvenOddBitwise{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number: ");
        int a=sc.nextInt();

        if((a&1)==0){
            System.out.println("number is Even.");
        }else{
            System.out.println("number is Odd.");
        }



    }

}

/*
5 in binary 0101
    0101
&   0001       (AND: 0 0= 0 ; 0 1= 0; 1 0= 0; 1 1= 1)
----------
    0001       (1 -> Odd)

2 in binary 0010
    0010
&   0001       (AND: 0 0= 0 ; 0 1= 0; 1 0= 0; 1 1= 1)
----------
    0000       (0 -> Odd)

*/
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac EvenOddBitwise.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java EvenOddBitwise
Enter the number:
5
number is Odd.

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

```java
/*
Q3:  Implement a program that calculates the  sum of digits  of an integer using  modulus
(  %  ) and division (  /  ) operators
*/
import java.util.Scanner;

public class SumOfDigits{

    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter number a=");
        int a=sc.nextInt();

        int sum=0;

        while(a != 0){
            sum=sum+a%10;
            a=a/10;

        }

        System.out.println("Sum of digit: "+sum);

        sc.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac SumOfDigits.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java SumOfDigits
Enter number a=
18
Sum of digit: 9

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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
/*
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
*/

public class DivisibilityByThree {
    public static boolean isDivisibleBy3(int num) {
        if (num < 0) num = -num; // Handle negative numbers

        while (num > 0) {
            int oddSum = 0, evenSum = 0;
            int position = 0;

            while (num > 0) {
                if ((num & 1) == 1) { // Check if the bit is 1
                    if (position % 2 == 0)
                        evenSum++;
                    else
                        oddSum++;
                }
                num = num >> 1; // Right shift to check next bit
                position++;
            }

            num = Math.abs(oddSum - evenSum); // Reduce the number
        }

        return num == 0;
    }

    public static void main(String[] args) {
        int num = 27; // Test number
        if (isDivisibleBy3(num))
            System.out.println(num + " is divisible by 3.");
        else
            System.out.println(num + " is not divisible by 3.");
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac DivisibilityByThree.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java DivisibilityByThree
27 is divisible by 3.

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

```java
/*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
*/

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

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

        // Swapping without using a third variable
        a += b; // a = a + b
        b = a - b; // b = (a + b) - b = a
        a -= b; // a = (a + b) - a = b

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

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac SwapNumbers1.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java SwapNumbers1
Before swapping: a = 5, b = 10
After swapping: a = 10, b = 5

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

2. Relational & Logical Operators

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

```
/*
Q6:  Write a program to find the  Largest of three numbers  using only the  ternary opera
(  ? :

*/
import java.util.Scanner;

public class LargestOfThree{

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

        System.out.println("Enter the number a= ");
        int a=sc.nextInt();
        System.out.println("Enter the number b= ");
        int b=sc.nextInt();
        System.out.println("Enter the number c= ");
        int c=sc.nextInt();

        int largest=(a>b)?((a>c)?a:c):((b>c)?b:c);

        System.out.println("The largest Number is: "+largest);


    }

}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac LargestOfThree.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java LargestOfThree
Enter the number a=
5
Enter the number b=
3
Enter the number c=
19
The largest Number is: 19

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

```java
/*Q7:  Implement a Java program that checks whether a given year is a  leap year or not  using
logical (  &&  ,  ||  ) operators
*/
import java.util.Scanner;

class LeapYearChecker{

    public static void main(String[] args){

    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a year: ");
    int year=sc.nextInt();

    // Leap year condition using logical operators
    boolean isLeap=(year %4 == 0 && year % 100 !=0)|| (year % 400 ==0);

    if(isLeap){
        System.out.println(year+" is a leap year");

    }else{
        System.out.println(year+" is not a leap year");
    }

    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac LeapYearChecker.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java LeapYearChecker
Enter a year:
2025
2025 is not a leap year

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

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

```java
/*
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 AtLeastTwoTrue{

    public static void main(String[] args){

        Scanner sc=new Scanner(System.in);
        System.out.println("Enter first boolean value (True/false):");
        boolean a=sc.nextBoolean();

        System.out.println("Enter second boolean value (True/false):");
        boolean b=sc.nextBoolean();

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

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

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

        sc.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac AtLeastTwoTrue.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java AtLeastTwoTrue
Enter first boolean value (True/false):
true
Enter second boolean value (True/false):
false
Enter third boolean value (True/false):
true
At least two inputs are true:true
```

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**.

```java
/*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
*/

import java.util.Scanner;

class RangeCheck{

    public static void main(String[] args){

        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number:");
        int a=sc.nextInt();

        boolean isInRange = (a >= 20 && a <= 50);

        System.out.println("is the number in the range 20 to 50?  : "+isInRange);

        sc.close();

    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac RangeCheck.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java RangeCheck
Enter the number:
28
is the number in the range 20 to 50?  : true
```

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

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

*/
import java.util.Scanner;

class VowelOrConsonant{

    public static void main(String[] args){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the character: ");
        char ch=sc.next().toLowerCase().charAt(0); // Convert to lowercase for easy comparison

        String result = (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 a: "+result);

        sc.close();

    }

}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac VowelOrConsonant.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java VowelOrConsonant
Enter the character:
r
The character r is a: Consonant

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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.

```java
/*
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
*/
import java.util.Scanner;

class PowerOfTwoCheck{

    public static void main(String[] main){

    Scanner sc=new Scanner(System.in);

    System.out.println("Enter the number: ");
    int n=sc.nextInt();

    boolean isPowerOfTwo=(n>0) && ((n &(n-1))==0);

    System.out.println(n+" is a power of 2: "+isPowerOfTwo);

    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac PowerOfTwoCheck.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java PowerOfTwoCheck
Enter the number:
8
8 is a power of 2: true

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

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

class MultiplyByEight{

    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number: ");
        int num=sc.nextInt();

        int result=num<<3; // Left shifting by 3 places (equivalent to multiplying by 8)

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

    }
}

/*

Bitwise Left Shift (<<):

The left shift operator (<<) shifts all bits to the left by a given number of positions.
Each left shift by 1 doubles the number.
Since 8 = 2³, shifting left by 3 (<< 3) multiplies the number by 8.

num << 3  →  num × 2³  →  num × 8

5 << 3  →  5 × 8  →  40
*/
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac MultiplyByEight.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java MultiplyByEight
Enter the number:
5
5 multiplied by 8 is: 40

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

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

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

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();
        scanner.close();

        int mask = num >> 31;  // Extract sign bit (0 for positive, -1 for negative)
        int abs = (num + mask) ^ mask;  // Compute absolute value

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

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac AbsoluteValue.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java AbsoluteValue
Enter an integer: -15
Absolute value: 15

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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)`.

```java
/*
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)

*/

import java.util.Scanner;

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

        int count = 0;
        while (n != 0) {
            n = n & (n - 1); // Clears the rightmost set bit
            count++;
        }

        System.out.println("Number of set bits: " + count);
    }
}

/*
n & (n - 1) removes the rightmost set bit in n.
We keep applying this operation until n becomes 0, counting how many times it runs.

Step-by-step process:
1101 & 1100 → 1100 (count = 1)
1100 & 1011 → 1000 (count = 2)
1000 & 0111 → 0000 (count = 3)
*/
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac CountSetBits.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java CountSetBits
Enter an integer: 19
Number of set bits: 3

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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`.

```java
/*
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

*/

import java.util.Scanner;

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

        // Mask to get even-positioned bits (bits at positions 0, 2, 4, ...)
        int evenBits = x & 0xAAAAAAAA;

        // Mask to get odd-positioned bits (bits at positions 1, 3, 5, ...)
        int oddBits = x & 0x55555555;

        // Right shift even bits
        evenBits >>= 1;

        // Left shift odd bits
        oddBits <<= 1;

        // Combine the swapped bits
        int swapped = evenBits | oddBits;

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


/*
```

```
/*

input x = 23 (Binary: 0001 0111)
Use bit masks:

0xAAAAAAAA (10101010 10101010 ...) selects even-positioned bits (index 0, 2, 4...).
0x55555555 (01010101 01010101 ...) selects odd-positioned bits (index 1, 3, 5...).

1] Extract even bits:
    Right shift-
    int evenBits = x & 0xAAAAAAAA;

    0001 0111
&   1010 1010
    ------------
    0000 0010  (even bits)

    After right shift: 0000 0001

2] Extract odd bits:
    Left shift-
     int oddBits = x & 0x55555555;

    0001 0111
&   0101 0101
    ------------
    0001 0101  (odd bits)

    After left shift: 0010 1010

3] combine both odd and even swapped bits. use the bitwise OR (|) operation.
    0000 0001
    0010 1010
    ------------
    0010 1011    result = 43
*/
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac SwapOddEvenBits.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java SwapOddEvenBits
Enter an integer: 23
Number after swapping odd and even bits: 43

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

4. Ternary Operator Challenges

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

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

*/


import java.util.Scanner;

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

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

        String result = (num > 0) ? "Positive" : (num < 0) ? "Negative" : "Zero";

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

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac NumberCheck.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java NumberCheck
Enter a number: 5
The number is: Positive

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

```java
/*
Q17:  Implement a Java program that finds the  minimum of four numbers  using nested
ternary operators
*/

import java.util.Scanner;

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

        System.out.print("Enter four numbers: ");
        int a = scanner.nextInt();
        int b = scanner.nextInt();
        int c = scanner.nextInt();
        int d = scanner.nextInt();

        // Using nested ternary operator to find the minimum
        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("The minimum number is: " + min);

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac MinOfFour.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java MinOfFour
Enter four numbers: 2
5
9
1
The minimum number is: 1

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

```java
/*
Q18:  Given a student's percentage, print  "Pass"  if the percentage is 40 or above;
otherwise, print  "Fail"  , using only the ternary operator.
*/

import java.util.Scanner;

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

        System.out.print("Enter the student's percentage: ");
        double percentage = scanner.nextDouble();

        // Using ternary operator to check pass/fail
        String result = (percentage >= 40) ? "Pass" : "Fail";

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

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac PassOrFail.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java PassOrFail
Enter the student's percentage: 75
Result: Pass

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

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

*/
import java.util.Scanner;

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

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

        // Using ternary operator to check character type
        String result = (ch >= 'A' && ch <= 'Z') ? "Uppercase" :
                        (ch >= 'a' && ch <= 'z') ? "Lowercase" : "Not a letter";

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

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac CharacterCheck.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java CharacterCheck
Enter a character: a
The character is: Lowercase

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

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

*/

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 num = scanner.nextInt();

        // Using ternary operator to find absolute value
        int absValue = (num < 0) ? -num : num;

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

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java AbsoluteValue1
Enter a number: 5
Absolute value: 5

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

5. Miscellaneous Operator Questions

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

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

import java.util.Scanner;

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

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

        // Increment using bitwise operations
        int incrementedNum = -~num; // Equivalent to num + 1

        System.out.println("Incremented number: " + incrementedNum);

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac IncrementWithoutPlus.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java IncrementWithoutPlus
Enter a number: 5
Incremented number: 6

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

```java
/*
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;

public class SwitchCalculator {
    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;

        // Switch-case for calculator operations
        switch (operator) {
            case '+':
                result = num1 + num2;
                System.out.println("Result: " + result);
                break;
            case '-':
                result = num1 - num2;
                System.out.println("Result: " + result);
                break;
            case '*':
                result = num1 * num2;
                System.out.println("Result: " + result);
                break;
            case '/':
                if (num2 != 0) {
                    result = num1 / num2;
                    System.out.println("Result: " + result);
                } else {
                    System.out.println("Error: Division by zero is not allowed.");
                }
                break;
            default:
                System.out.println("Invalid operator! Please use +, -, *, or /.");
        }

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac SwitchCalculator.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java SwitchCalculator
Enter first number: 5
Enter an operator (+, -, *, /): +
Enter second number: 7
Result: 12.0

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

```java
/*
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;

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

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

        // Using bitwise AND to check odd/even and printing directly
        System.out.println("The number is: " + ((num & 1) == 0 ? "Even" : "Odd"));

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac OddEvenBitwise.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java OddEvenBitwise
Enter a number: 5
The number is: Odd

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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

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

public class EvenNumbersBitwise {
    public static void main(String[] args) {
        System.out.println("Even numbers from 1 to 100:");

        for (int i = 1; i <= 100; i++) {
            // Using bitwise AND to check even numbers
            if ((i & 1) == 0) {
                System.out.print(i + " ");
            }
        }
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac EvenNumbersBitwise.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java EvenNumbersBitwise
Even numbers from 1 to 100:
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 8
4 86 88 90 92 94 96 98 100
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```

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; }`

```java
/*
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;

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

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

        int rev = 0;

        while (num != 0) {
            rev = rev * 10 + num % 10; // Extract last digit and add to rev
            num /= 10; // Remove last digit from num
        }

        System.out.println("Reversed number: " + rev);

        scanner.close();
    }
}
```

```
D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>javac ReverseInteger.java

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>java ReverseInteger.java
Enter an integer: 123
Reversed number: 321

D:\PG-DAC\OOPS\JavaProgram\Assignment\Assignment-2>
```