1. **Implement a Java program to swap two variables without using a third variable.**

**Example Input:**

int a = 10, b = 20;

**Expected Output:**

Before Swap: a = 10, b = 20

After Swap: a = 20, b = 10

1. **Write a program that demonstrates type promotion in expressions involving mixed data types (e.g., int, float, double, char).**

**Example Input:**

int a = 5;

char b = 'A';

double c = 2.5;

**Expected Output:**

Result of (a + b \* c) = <calculated\_value>

Explain how Java promotes types in the expression.

1. **Write a program to reverse the bits of an integer.**

**Example Input:**

int num = 5; // Binary: 0000 0101

**Expected Output:**

Reversed Binary: 1010 0000 (Equivalent Decimal: 160)

1. **Write a program to check whether a number is power of two using bitwise operators.**

**Example Input/Output:**

Input: 16

Output: Yes, it is a power of two

Input: 18

Output: No, it is not a power of two

1. **Write a Java program to evaluate a mathematical expression given as a string and return the result.**

**Example Input:**

Input: "10 + 2 \* 6"

Output: 22

Input: "100 \* (2 + 12) / 14"

Output: 100

1. **Implement a Java program to find the sum of digits of a given number using a while loop.**

**Example Input/Output:**

Input: 1234

Output: 10

(Explanation: 1 + 2 + 3 + 4 = 10)

1. **Implement a Java program that checks whether a given number is a prime number using a for loop.**

**Example Input/Output:**

Input: 13

Output: Yes, 13 is a prime number

Input: 20

Output: No, 20 is not a prime number

1. **Implement a program to generate the first N Fibonacci numbers using a do-while loop.**

**Example Input/Output:**

Input: 5

Output: 0, 1, 1, 2, 3

1. **Write a program to print a pattern using nested loops.**

**Example Input/Output:**

Input: 5

Output:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

(The pattern should be a **pyramid of stars** based on the given input.)

1. Your application requires a **custom immutable class** to store configurations.

Implement an **immutable** Configuration class that stores databaseUrl, username, and password.

Ensure **no setter methods** exist, and values cannot be changed once assigned.

**Example Usage:**

Configuration config = new Configuration("jdbc:mysql://localhost", "admin", "password123");

System.out.println(config.getDatabaseUrl());

1. You are designing an **image processing tool** that needs a function to **convert RGB color values to grayscale** using bitwise operations.

Write a method that takes (red, green, blue) values (0-255) and converts them into a **single grayscale value** using the formula: grayscale=0.3R+0.59G+0.11Bgrayscale = 0.3R + 0.59G + 0.11Bgrayscale=0.3R+0.59G+0.11B

Ensure efficient calculations using **bitwise operations**.

1. A software system needs an **integer compression technique** to store large numbers efficiently.

Implement a method to **encode an integer** using bitwise operations.

Write another method to **decode the integer** back.

**Example Usage:**

int encoded = encode(12345);

int decoded = decode(encoded);

System.out.println(decoded); // Output: 12345

1. Your client wants a **discount calculator** for an online store with the following rules:

If the cart value is **above $500**, apply a **20% discount**.

If the cart value is **between $100 and $500**, apply a **10% discount**.

Otherwise, no discount.

Write a function calculateFinalPrice(double cartValue) that returns the **final amount** after applying the discount.

1. A banking system requires a **custom ATM withdrawal** function:

Implement a method that **dispenses cash in denominations of 100, 500, and 2000** rupees, minimizing the number of notes.

Ensure proper error handling if the amount is not a multiple of 100.

**Example Input/Output:**

Input: 3700

Output: 1 x 2000, 3 x 500, 2 x 100

1. You are given a number **N** as input. Write a Java program that prints the multiplication table of **N** using **all three types of loops (for, while, do-while)**. Additionally, modify the program to:

Print the multiplication table **in reverse order** (from N x 10 to N x 1) using a for loop.

Allow the user to specify a **custom range** (e.g., from start to end) for the multiplication table using a while loop.

Ensure the program asks the user if they want to generate another table and continues until the user chooses to stop (use a do-while loop).