RPS DAY 12 Assignments

Assignment 5

Name: Akshada Baad
Batch - CPPE

Day 12:

Task 1: Bit Manipulation Basics

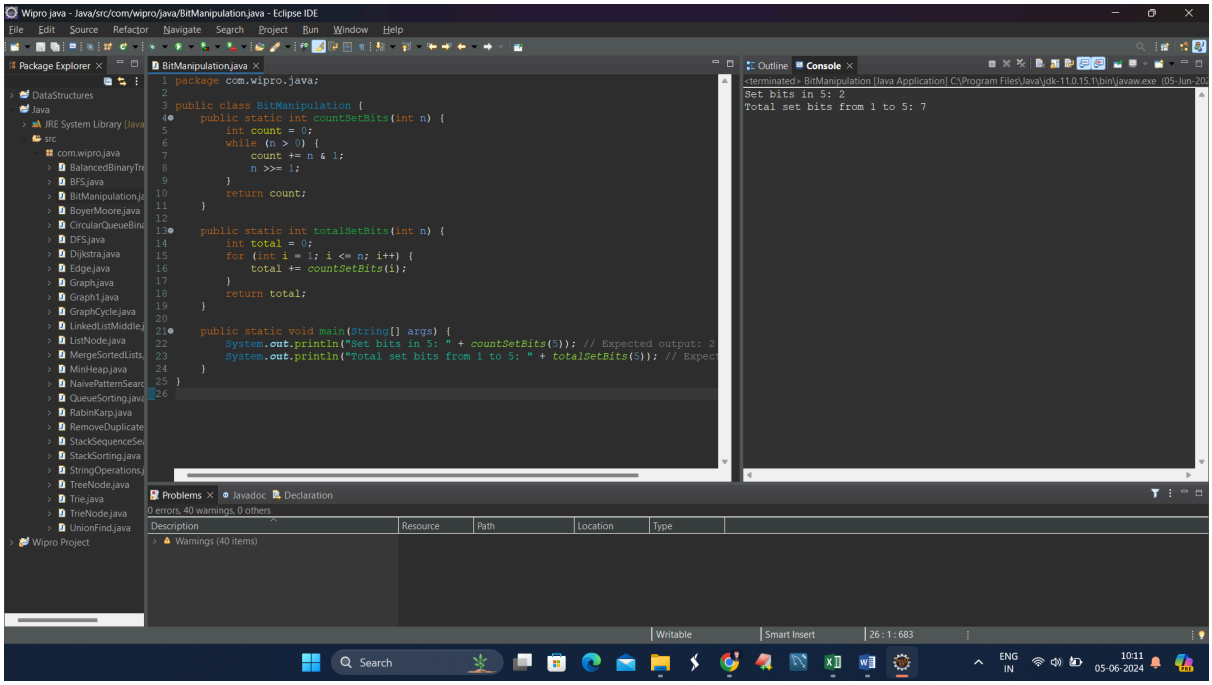
Create a function that counts the number of set bits (1s) in the binary representation of an integer. Extend this to count the total number of set bits in all integers from 1 to n.

```
 public class BitManipulation {
public static int countSetBits(int n) {
    int count = 0;
    while (n > 0) {
        count += n & 1;
        n >>= 1;
    }
    return count;
}

public static int totalSetBits(int n) {
    int total = 0;
    for (int i = 1; i <= n; i++) {
        total += countSetBits(i);
    }
    return total;
}

public static void main(String[] args) {
    System.out.println("Set bits in 5: " + countSetBits(5)); // Expected output: 2
```

```
        System.out.println("Total set bits from 1 to 5: " + totalSetBits(5)); // Expected output: 7
    }
}
```



Task 2: Unique Elements Identification

Given an array of integers where every element appears twice except for two, write a function that efficiently finds these two non-repeating elements using bitwise XOR operations.

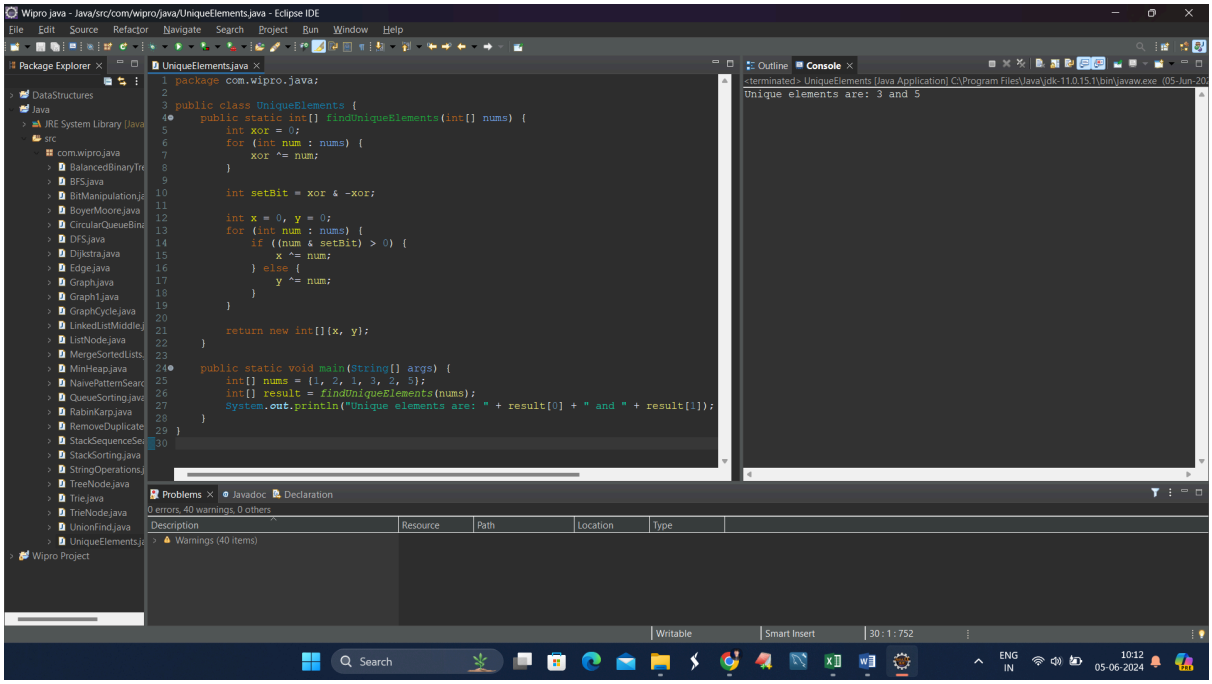
```
public class UniqueElements {
    public static int[] findUniqueElements(int[] nums) {
        int xor = 0;
        for (int num : nums) {
            xor ^= num;
        }

        int setBit = xor & -xor;
```

```
        int x = 0, y = 0;
        for (int num : nums) {
            if ((num & setBit) > 0) {
                x ^= num;
            } else {
                y ^= num;
            }
        }

        return new int[]{x, y};
    }

    public static void main(String[] args) {
        int[] nums = {1, 2, 1, 3, 2, 5};
        int[] result = findUniqueElements(nums);
        System.out.println("Unique elements are: " + result[0] + " and " + result[1]); // Expected output: 3 and 5
    }
}
```




```java
package com.wipro.java;

public class UniqueElements {
    public static int[] findUniqueElements(int[] nums) {
        int xor = 0;
        for (int num : nums) {
            xor ^= num;
        }

        int setBit = xor & -xor;

        int x = 0, y = 0;
        for (int num : nums) {
            if ((num & setBit) > 0) {
                x ^= num;
            } else {
                y ^= num;
            }
        }

        return new int[]{x, y};
    }

    public static void main(String[] args) {
        int[] nums = {1, 2, 1, 3, 2, 5};
        int[] result = findUniqueElements(nums);
        System.out.println("Unique elements are: " + result[0] + " and " + result[1]);
    }
}
```