Day - 12

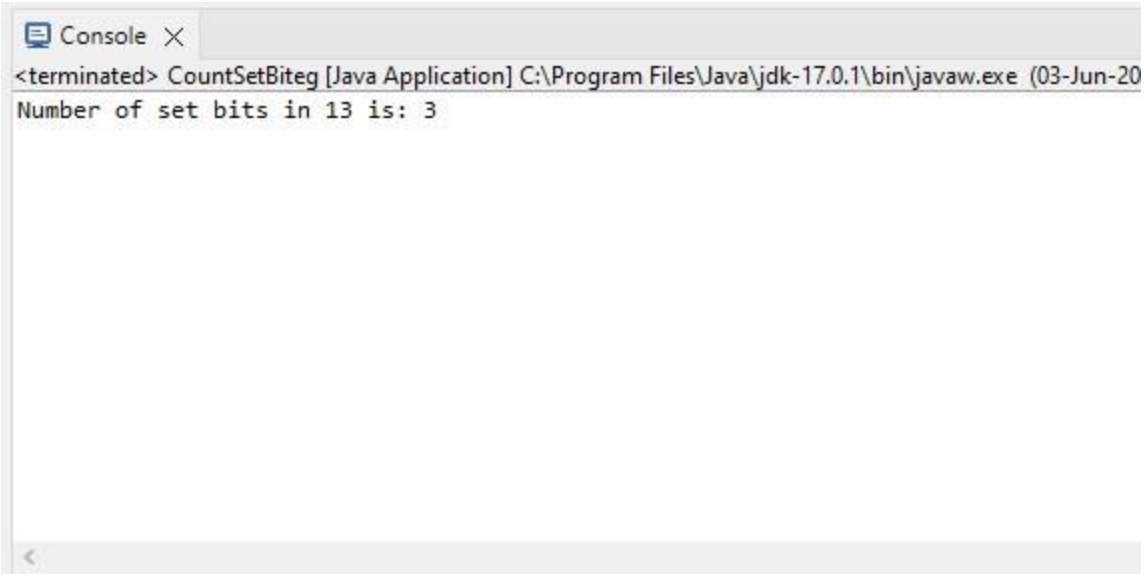
1] Task 1: Bit Manipulation Basics
Create a function that counts the number of set bits (1s) in the binary representation of an integer. Extend this to count the total number of set bits in all integers from 1 to n.

Solution:-
Code -

CountSetBiteg.java

```java
package com.wipro.graphalgo;

public class CountSetBiteg {


        public static int countSetBits(int n) {
            int count = 0;
            while (n > 0) {
                n &= (n - 1);
                count++;

            }
            return count;
        }

        public static void main(String[] args) {
            int num = 13;
            System.out.println("Number of set bits in " + num + " is: " + countSetBits(num));
        }
    }
```

Output-



2] 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.

Solution:-
Code -

UniqueElements.java

```java
package com.wipro.graphalgo;

public class UniqueElements {



	public static int[] findTwoNonRepeatingElements(int[] nums) {

		int xor = 0;
		for (int num : nums) {
			xor ^= num;
		}


		int rightmostSetBit = xor & ~(xor-1);
		int x = 0, y = 0;
		for (int num : nums) {
			if ((num & rightmostSetBit) >0) {
				x ^= num;
			} else {
				y ^= num;
			}
		}

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

	public static void main(String[] args) {
		int[] nums = {2,4,7,9,2,4,5,7};
		int[] result = findTwoNonRepeatingElements(nums);
		System.out.println("The two non-repeating elements are: " + result[0] + " and " + result[1]);
	}
}

```

Output -

Console

<terminated> UniqueElements [Java Application] C:\Program Files\Java\jdk

```
The two non-repeating elements are: 5 and 9
```