

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 -

```
CountSetBits.java X
1 package com.wipro.graphalgo;
2
3 public class CountSetBits {
4
5
6     public static int countSetBits(int n) {
7         int count = 0;
8         while (n > 0) {
9             n &= (n - 1);
10            count++;
11        }
12        return count;
13    }
14
15    public static void main(String[] args) {
16        int num = 13;
17        System.out.println("Number of set bits in " + num + " is: " + countSetBits(num));
18    }
19
20 }
21
22
23 |
```

Output-

A screenshot of a Java console window. The title bar shows a 'Console' tab with a close button. The text inside the console reads: '<terminated> CountSetBiteg [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (03-Jun-20' followed by a new line and 'Number of set bits in 13 is: 3'.

```
<terminated> CountSetBiteg [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (03-Jun-20
Number of set bits in 13 is: 3
```

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 ×
1 package com.wipro.graphalgo;
2
3 public class UniqueElements {
4
5
6
7     public static int[] findTwoNonRepeatingElements(int[] nums) {
8
9         int xor = 0;
10        for (int num : nums) {
11            xor ^= num;
12        }
13
14        int rightmostSetBit = xor & ~(xor-1);
15        int x = 0, y = 0;
16        for (int num : nums) {
17            if ((num & rightmostSetBit) > 0) {
18                x ^= num;
19            } else {
20                y ^= num;
21            }
22        }
23
24        return new int[]{x, y};
25    }
26
27    public static void main(String[] args) {
28        int[] nums = {2,4,7,9,2,4,5,7};
29        int[] result = findTwoNonRepeatingElements(nums);
30        System.out.println("The two non-repeating elements are: " + result[0] + " and " + result[1]);
31    }
32 }
33
34

```

Output -

```

Console ×
<terminated> UniqueElements [Java Application] C:\Program Files\Java\jdk
The two non-repeating elements are: 5 and 9

```