

## 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.**

```
package com.wipro.patterns;
```

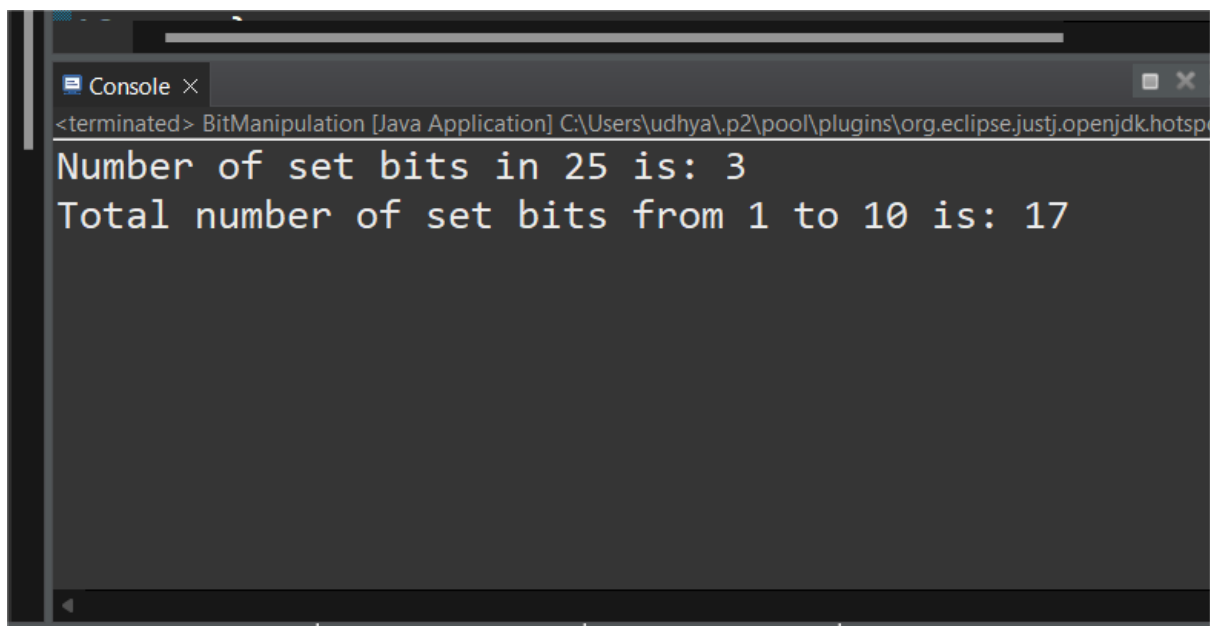
```
public class BitManipulation {
```

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

```
    public static int countTotalSetBits(int n) {  
        int totalCount = 0;  
        for (int i = 1; i <= n; i++) {  
            totalCount += countSetBits(i);  
        }  
        return totalCount;  
    }
```

```
public static void main(String[] args) {  
    int num = 25;  
  
    System.out.println("Number of set bits in " + num + " is: " +  
countSetBits(num));  
  
    int n = 10;  
  
    System.out.println("Total number of set bits from 1 to " + n + " is: " +  
countTotalSetBits(n));  
}  
}
```

## OUTPUT:

A screenshot of a Java IDE's console window. The window title is "Console x". The text inside the console shows the output of the program: "Number of set bits in 25 is: 3" followed by "Total number of set bits from 1 to 10 is: 17". The background of the console is dark with light-colored text.

```
<terminated> BitManipulation [Java Application] C:\Users\udhya\p2\pool\plugins\org.eclipse.justj.openjdk.hotsp  
Number of set bits in 25 is: 3  
Total number of set bits from 1 to 10 is: 17
```

## 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.**

```
package com.wipro.patterns;
```

```
public class XOR {

    public static void findNonRepeatingElements(int[] nums) {
        int xor = 0;

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

        int rightmostSetBit = xor & -xor;

        int firstUnique = 0;
        int secondUnique = 0;

        for (int num : nums) {
            if ((num & rightmostSetBit) != 0) {

                firstUnique ^= num;
            } else {

                secondUnique ^= num;
            }
        }
    }
}
```

```
        System.out.println("The two non-repeating elements are: " +  
firstUnique + " and " + secondUnique);
```

```
    }
```

```
    public static void main(String[] args) {
```

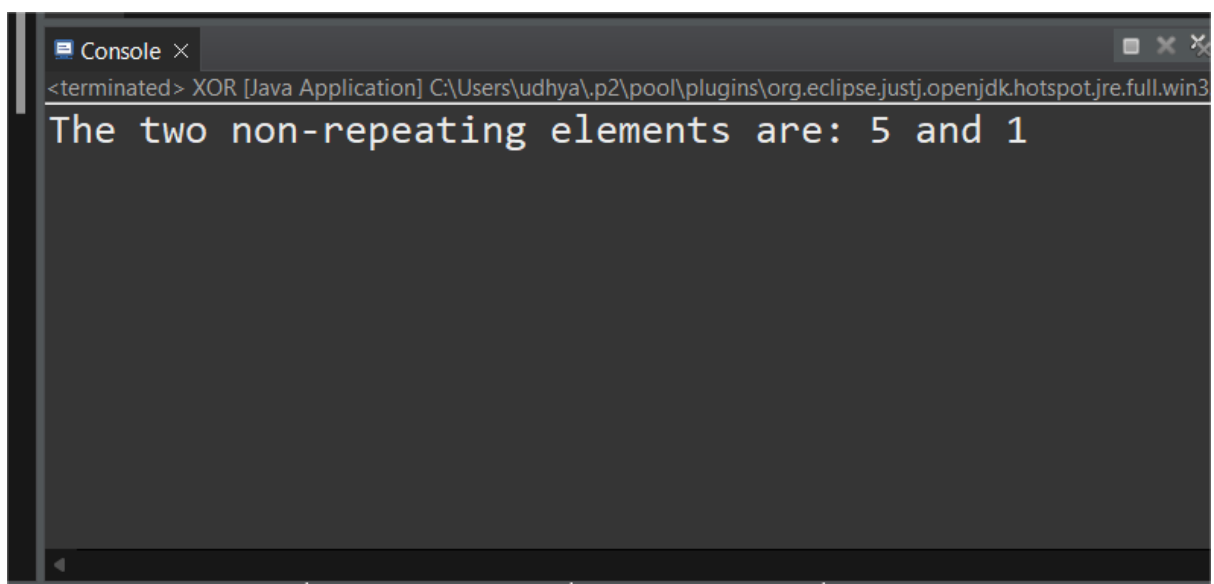
```
        int[] nums = {4, 2, 4, 5, 2, 3, 3, 1};
```

```
        findNonRepeatingElements(nums);
```

```
    }
```

```
}
```

## OUTPUT:

A screenshot of a Java console window. The title bar shows 'Console' with a close button. The text inside the console reads: '<terminated> XOR [Java Application] C:\Users\udhya\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win3'. Below this, the output of the program is displayed: 'The two non-repeating elements are: 5 and 1'. The console has a dark background and a light-colored text color. There are standard window controls (minimize, maximize, close) in the top right corner.

```
<terminated> XOR [Java Application] C:\Users\udhya\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win3  
The two non-repeating elements are: 5 and 1
```