Assignment – 15

Prakash Manikanta Irrinki

Prakashnaidu9494@gmail.com

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.

Program:

```java
public class BitManuplatioin {

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

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

    public static void main(String[] args) {
        int num=10;
        System.out.println("Count of set bits in binary
                            representation of "+num+" is:
                            "+countSetBIts(num));
        System.out.println("Total count of set bits in binary
                            representation of 1 to "+num+" is:
                            "+totalSetBits(num));
    }

}
```

Output:

```
Count of set bits in binary representation of 10 is: 2
Total count of set bits in binary representation of 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.

Program:

```java
public class UniqueElements {

    public static int findRightMostBit(int num) {
        int position = 0;
        while ((num & 1) == 0) {
            position++;
            num >>= 1;
        }
        return position;
    }

    public static void findNonRepetingElements(int[] arr) {
        int result = 0;
        for (int num : arr) {
            result ^= num;
        }
        int rightMostSetBit = findRightMostBit(result);

        int group1 = 0;
        int group2 = 0;
        for (int num : arr) {
            if (((num >> rightMostSetBit) & 1) == 0) {
                group1 ^= num;
            } else {
                group2 ^= num;
            }
        }
        System.out.println("Non repeating are: " + group1 +
                                 " and " + group2);
    }

    public static void main(String[] args) {
        int[] arr = { 2, 4, 5, 9, 2, 4 };
        findNonRepetingElements(arr);

    }

}
```

Output:

```
Non repeating are: 9 and 5
```