

Name :Tejaswini Anil Kamble

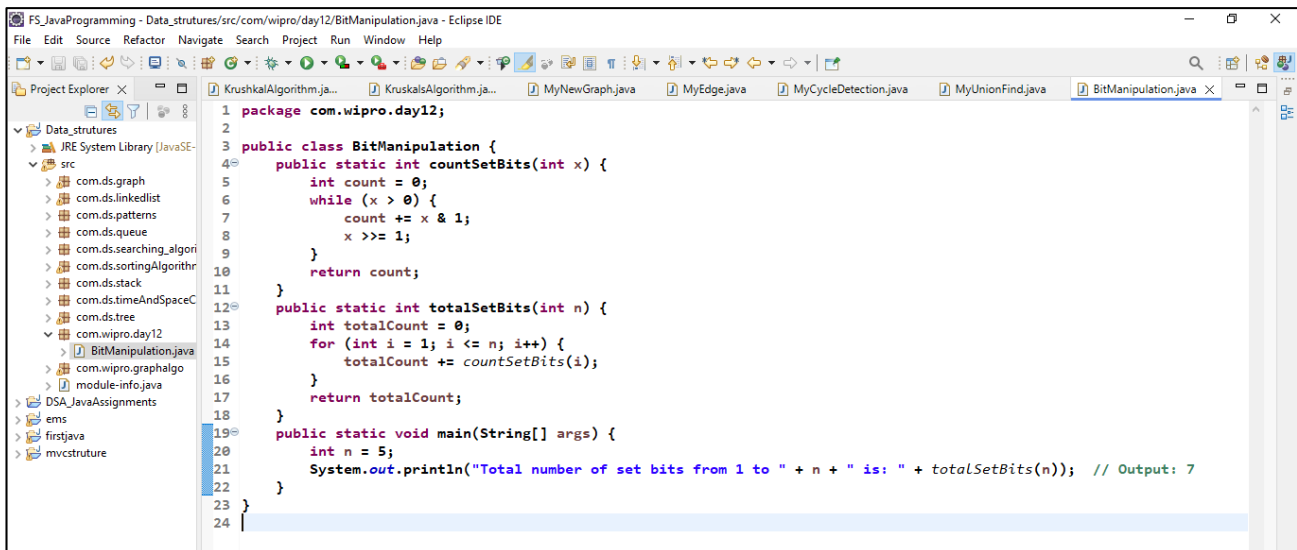
Email: teju000kamble@gmail.com

DAY 12 : Assignments

Task 1: Bit Manipulation Basics

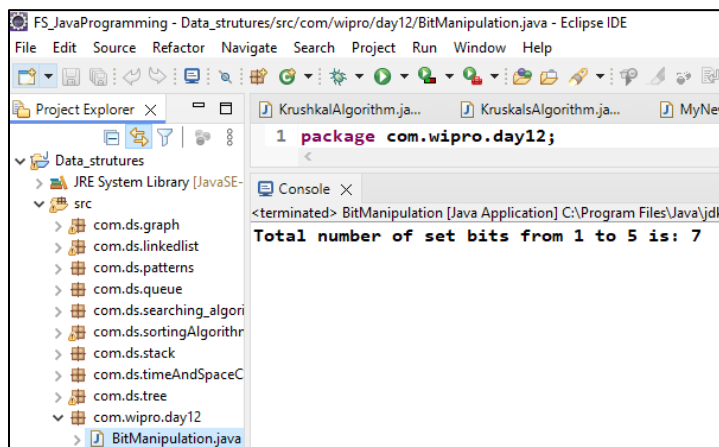
Create a function that counts the number of set bits (1s) in the binary representation of an integer. Extend this to count the total number of set bits in all integers from 1 to n.

Ans: Source code



```
1 package com.wipro.day12;
2
3 public class BitManipulation {
4     public static int countSetBits(int x) {
5         int count = 0;
6         while (x > 0) {
7             count += x & 1;
8             x >>= 1;
9         }
10        return count;
11    }
12    public static int totalSetBits(int n) {
13        int totalCount = 0;
14        for (int i = 1; i <= n; i++) {
15            totalCount += countSetBits(i);
16        }
17        return totalCount;
18    }
19    public static void main(String[] args) {
20        int n = 5;
21        System.out.println("Total number of set bits from 1 to " + n + " is: " + totalSetBits(n)); // Output: 7
22    }
23 }
24
```

Output:

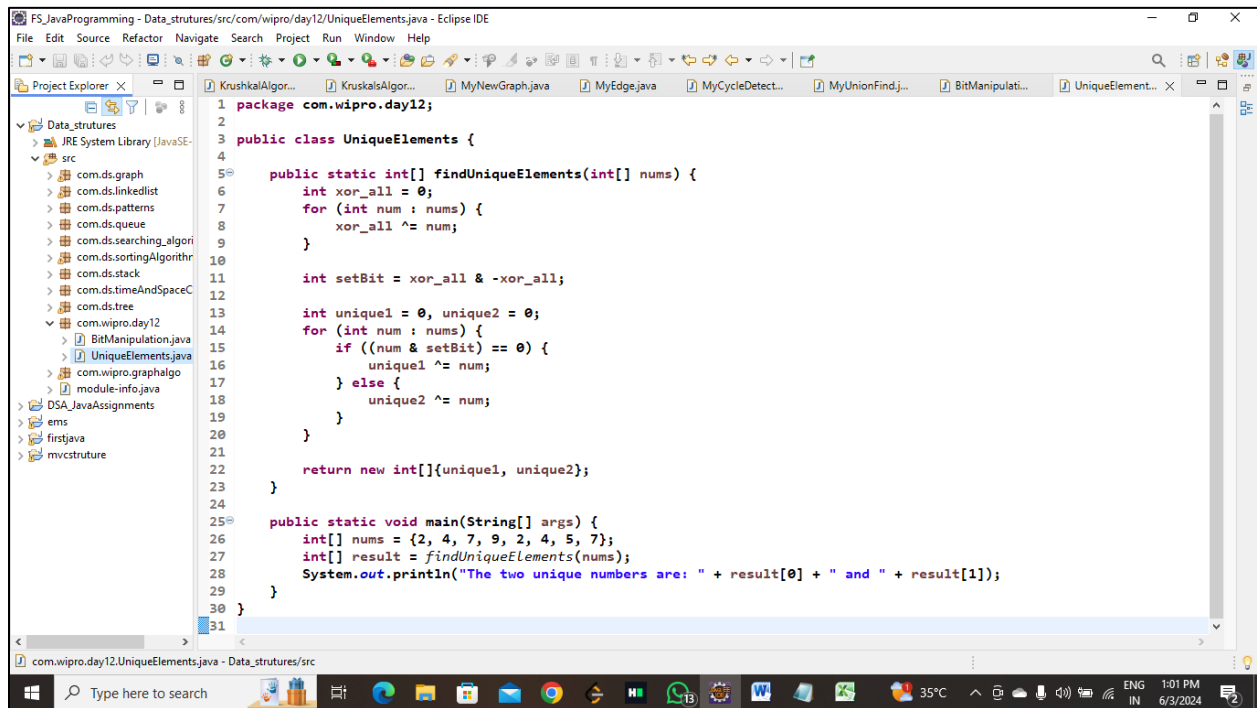


```
1 package com.wipro.day12;
2
3 public class BitManipulation {
4     public static int countSetBits(int x) {
5         int count = 0;
6         while (x > 0) {
7             count += x & 1;
8             x >>= 1;
9         }
10        return count;
11    }
12    public static int totalSetBits(int n) {
13        int totalCount = 0;
14        for (int i = 1; i <= n; i++) {
15            totalCount += countSetBits(i);
16        }
17        return totalCount;
18    }
19    public static void main(String[] args) {
20        int n = 5;
21        System.out.println("Total number of set bits from 1 to 5 is: 7");
22    }
23 }
24
```

Task 2: Unique Elements Identification

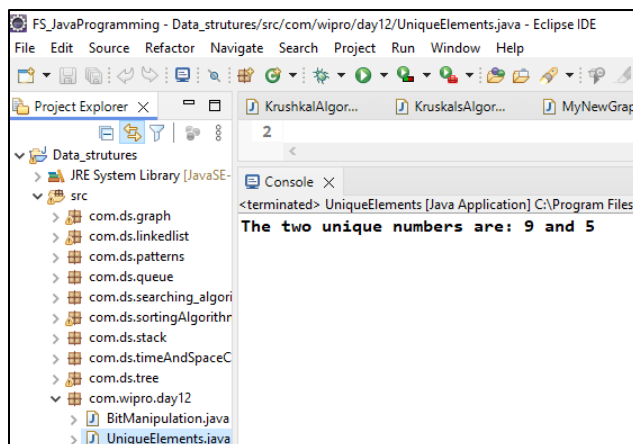
Given an array of integers where every element appears twice except for two, write a function that efficiently finds these two non-repeating elements using bitwise XOR operations.

Ans: Source code



```
1 package com.wipro.day12;
2
3 public class UniqueElements {
4
5     public static int[] findUniqueElements(int[] nums) {
6         int xor_all = 0;
7         for (int num : nums) {
8             xor_all ^= num;
9         }
10
11         int setBit = xor_all & -xor_all;
12
13         int unique1 = 0, unique2 = 0;
14         for (int num : nums) {
15             if ((num & setBit) == 0) {
16                 unique1 ^= num;
17             } else {
18                 unique2 ^= num;
19             }
20         }
21
22         return new int[]{unique1, unique2};
23     }
24
25     public static void main(String[] args) {
26         int[] nums = {2, 4, 7, 9, 2, 4, 5, 7};
27         int[] result = findUniqueElements(nums);
28         System.out.println("The two unique numbers are: " + result[0] + " and " + result[1]);
29     }
30 }
31
```

Output:



```
<terminated> UniqueElements [Java Application] C:\Program Files\
The two unique numbers are: 9 and 5
```

