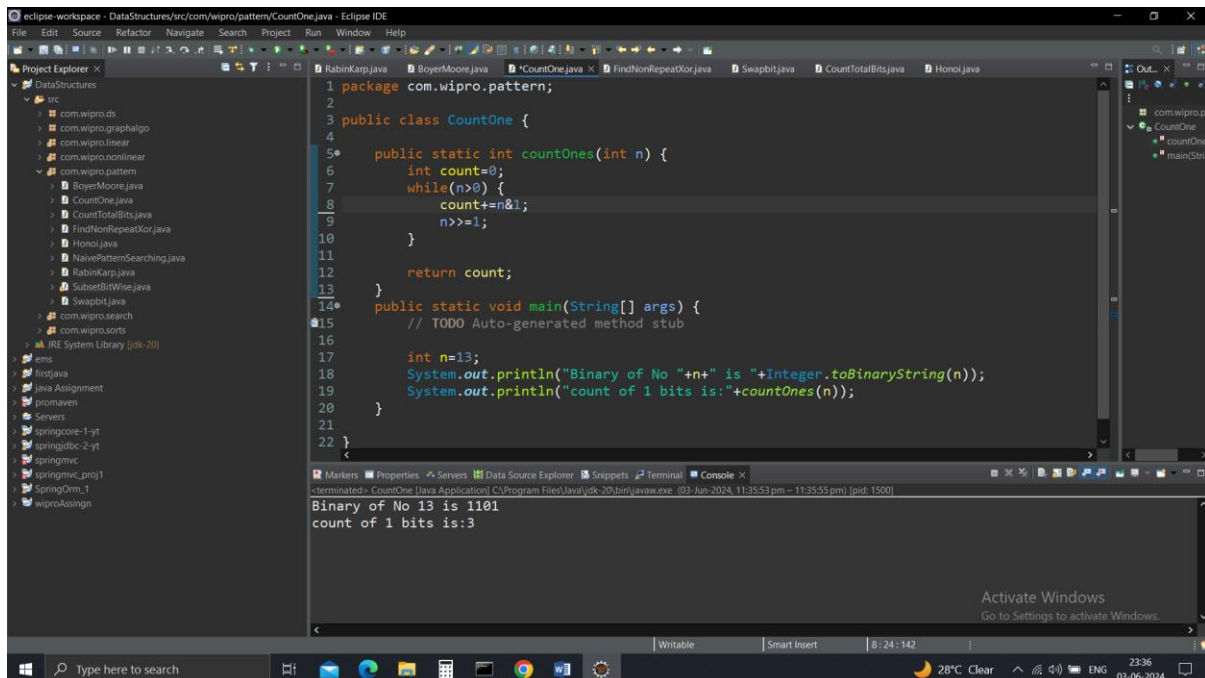


Task 1: Bit Manipulation Basics

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

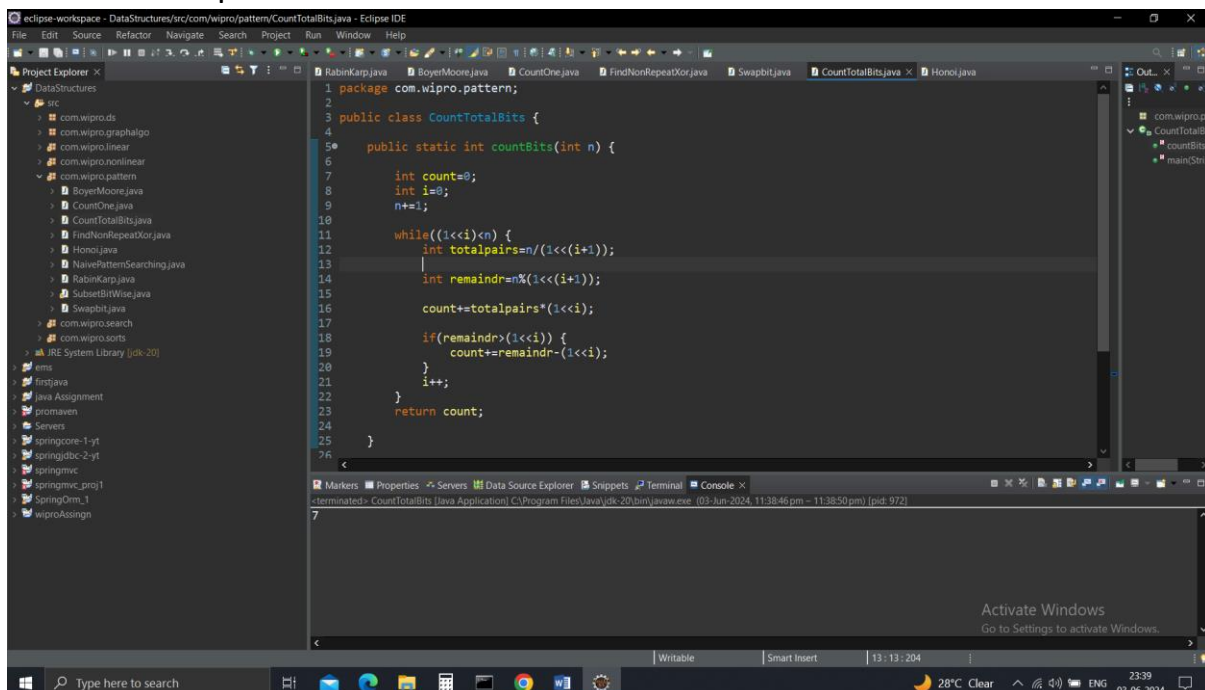


```
1 package com.wipro.pattern;
2
3 public class CountOne {
4
5     public static int countOnes(int n) {
6         int count=0;
7         while(n>0) {
8             count+=n&1;
9             n>>=1;
10        }
11
12        return count;
13    }
14
15    public static void main(String[] args) {
16        // TODO Auto-generated method stub
17
18        int n=13;
19        System.out.println("Binary of No "+n+" is "+Integer.toBinaryString(n));
20        System.out.println("count of 1 bits is:"+countOnes(n));
21    }
22 }
```

Binary of No 13 is 1101
count of 1 bits is:3

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.



```
1 package com.wipro.pattern;
2
3 public class CountTotalBits {
4
5     public static int countBits(int n) {
6
7         int count=0;
8         int i=0;
9         n+=1;
10
11         while((1<<i)<n) {
12             int totalpairs=n/(1<<(i+1));
13             int remaindr=n%(1<<(i+1));
14
15             count+=totalpairs*(1<<i);
16
17             if(remaindr>(1<<i)) {
18                 count+=remaindr-(1<<i);
19             }
20             i++;
21         }
22         return count;
23     }
24 }
25
26
27
```

7