Assignment Day-12

Core Java with DS and Algorithms

Name: Joshnitha Rangolu

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 day_12;

public class SetBitCount {

public static int countSetBits(int n) {

int count = 0;

while (n != 0) {

count += n & 1;

n >>= 1;

}

return count;

}

public static int countTotalSetBits(int n) {

int totalSetBits = 0;

for (int i = 1; i <= n; i++) {

totalSetBits += countSetBits(i);

}

return totalSetBits;

}

public static void main(String[] args) {

int num = 50;
```

```
int setBits = countSetBits(num);

System.out.println("Number of set bits in " + num + ": " + setBits);

int totalSetBits = countTotalSetBits(num);

System.out.println("Total set bits from 1 to " + num + ": " +
totalSetBits);

}

}
```

```
1  package day_12;
2
3  public class SetBitCount {
4      public static int countSetBits(int n) {
5          int count = 0;
6          while (n != 0) {
7            count += n & 1;
8            n >>= 1;
9          }
10         return count;
11       }
12
13       public static int countTotalSetBits(int n) {
14         int totalSetBits = 0;
15         for (int i = 1; i <= n; i++) {
16           totalSetBits += countSetBits(i);
17         }
18         return totalSetBits;
19       }
20
```

```
Console ×
<terminated> SetBitCount [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204
Number of set bits in 50: 3
Total set bits from 1 to 50: 136
```

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 day_12;

public class UniqueElements {

public static void findUniqueElements(int[] arr) {

int xor = arr[0];

for (int i = 1; i < arr.length; i++) {

xor ^= arr[i];

}
```

```
int rightmostSetBit = xor & ~(xor - 1);

int x = 0, y = 0;

for (int num : arr) {

if ((num & rightmostSetBit) != 0) {

x ^= num;

} else {

y ^= num;

}

}

System.out.println("Unique element 1: " + x);

System.out.println("Unique element 2: " + y);

}

public static void main(String[] args) {

int[] arr = {1, 2, 1, 3, 2, 4};

findUniqueElements(arr);

}

}
```