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.

SOLUTION :

```
package com.dsassignment_day_12;

public class CountBits {

public static int countTotalSetBits(int n) {

int total = 0;

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

total += countSetBits(i);

}

return total;

}

public static int countSetBits(int n) {

int count = 0;

while (n > 0) {

count += n & 1;

n >>= 1;

}

return count;

}

public static void main(String[] args) {

int n = 10;

System.out.println("Total set bits in all integers from 1 to " + n + " is:
" + countTotalSetBits(n));

}

}
```

OUTPUT:

```
Total set bits in all integers 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.

SOLUTION :

```
package com.dsassignment_day_12;

public class PrintNonRepeating {

static int[] findNonRepeating(int arr[]) {

int xor = 0;

int x = 0;

int y = 0;

for (int num : arr)

xor ^= num;

int set_bit_no = xor & ~(xor - 1);

for (int num : arr) {

if ((num & set_bit_no) > 0)

x ^= num;

else

y ^= num;

}

return new int[]{x, y};
```

```
}

public static void main(String[] args) {

int arr[] = {2, 4, 7, 9, 2, 4, 5, 7};

int[] result = findNonRepeating(arr);

System.out.println("The non-repeating elements are " + result[0] + " and "
+ result[1]);

}

}
```

OUTPUT: