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:

```
package com.Day18;
public class UniqueElements {
  public static int[] findUniqueElements(int[] nums) {
     // Step 1: XOR all elements to get XOR of the two unique
numbers
     int xor = 0;
     for (int num : nums) {
        xor ^= num;
     }
     // Step 2: Find a set bit in the XOR (rightmost set bit)
     int rightmostSetBit = xor & -xor;
     // Step 3: Divide elements into two groups and XOR within each
group
     int unique1 = 0, unique2 = 0;
     for (int num : nums) {
        if ((num & rightmostSetBit) == 0) {
           unique1 ^= num;
        } else {
           unique2 ^= num;
        }
     }
     return new int[]{unique1, unique2};
  }
  public static void main(String[] args) {
     int[] nums = {1, 2, 1, 3, 2, 5};
     int[] uniqueElements = findUniqueElements(nums);
     System.out.println("The two unique elements are: " +
uniqueElements[0] + " and " + uniqueElements[1]);
  }
}
```

OUTPUT:

```
The two unique elements are: 5 and 3
```