Leetcode

Leetcode Solution 1:-

```
class Solution {
    public int subarrayLCM(int[] nums, int k) {
        int n = nums.length;
        int ans = 0;
        for (int i = 0; i < n; ++i) {
            int a = nums[i];
            for (int j = i; j < n; ++j) {
                int b = nums[j];
                int x = lcm(a, b);
                if (x == k) {
                    ++ans;
                }
                a = x;
            }
        }
        return ans;
    }

    private int lcm(int a, int b) {
        return a * b / gcd(a, b);
    }

    private int gcd(int a, int b) {
        return b == 0 ? a : gcd(b, a % b);
    }
}
```

Leetcode Solution 2:-

```
class Solution {
    public int[] arrayChange(int[] nums, int[][]
operations) {
        Map<Integer, Integer> d = new
HashMap<>();
        for (int i = 0; i < nums.length; ++i) {
            d.put(nums[i], i);
        }
        for (var op : operations) {
            int a = op[0], b = op[1];
            nums[d.get(a)] = b;
            d.put(b, d.get(a));
        }
        return nums;
    }
}
```