



PROBLEM SOLVING

LeetCode



Two Sum

Problem Description:

<https://leetcode.com/problems/two-sum/>

Solution:

```
class Solution {
    List<int> twoSum(List<int> nums, int target) {
        // create a Map to store the complement of each number
        var numsMap = <int, int>{};

        // iterate through the list
        for (var i = 0; i < nums.length; i++) {
            // calculate the complement of the current element
            var complement = target - nums[i];

            // if the complement is already in the map, return the
indices
            if (numsMap.containsKey(complement)) {
                return [numsMap[complement]!, i];
            }

            // otherwise, add the current number and its index to the
map
            numsMap[nums[i]] = i;
        }

        // if no solution is found, return an empty list
        return [];
    }
}
```

Palindrome Number

Problem Description:

<https://leetcode.com/problems/palindrome-number/>

Solution:

```
class Solution {
    bool isPalindrome(int x) {
        return x.toString().split('').reversed.join() == x.toString();
    }
}
```

Roman to Integer

Problem Description:

<https://leetcode.com/problems/roman-to-integer/description/>

Solution:

```
class Solution {
    int romanToInt(String s) {
        Map<String, int> romanSymbols = {
            'I': 1,
            'V': 5,
            'X': 10,
            'L': 50,
            'C': 100,
            'D': 500,
            'M': 1000
        };

        int result = 0;

        for (int i = 0; i < s.length; i++) {
            if (i > 0 && romanSymbols[s[i]]! > romanSymbols[s[i - 1]]!)
            {
                result += romanSymbols[s[i]]! - 2 * romanSymbols[s[i -
1]]!;
            } else {
                result += romanSymbols[s[i]]!;
            }
        }

        return result;
    }
}
```


