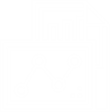
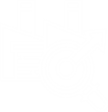
A picture containing purple, lilac, screenshot, violet

Description automatically generated­

LeetCode

SOLVING

PROBLEM

## Two Sum

Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.

**Example 1:**

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

**Example 2:**

Input: nums = [3,2,4], target = 6

Output: [1,2]

**Example 3:**

Input: nums = [3,3], target = 6

Output: [0,1]

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 [];

  }

}