1. Two Sum

Solved

Easy | Topics | Companies | Hint

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]
```

Code

C | Auto

```c

int* twoSum(int* nums, int numsSize, int target, int* returnSize) {
    // Allocate memory for result
    int* result = (int*)malloc(2 * sizeof(int));

    for (int i = 0; i < numsSize; i++) {
        for (int j = i + 1; j < numsSize; j++) {
            if (nums[i] + nums[j] == target) {
                result[0] = i;
                result[1] = j;
                *returnSize = 2;
                return result;
            }
        }
    }

    // As per problem, exactly one solution exists
    *returnSize = 0;
    return NULL;
}

```

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =

[2,7,11,15]

target =

9

Output

[0,1]

Expected

[0,1]

Contribute a testcase