1. Two Sum

▼ Link to problem on leetcode.

> Two Sum - LeetCode
>
> Can you solve this real interview question? Two Sum - Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to
>
> https://leetcode.com/problems/two-sum/

Problem Statement: “Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.”

Solution in C:

```
/*
  * Brute Force Method.
 */
int* twoSum(int* nums, int numsSize, int target, int* returnSize) {
    for (int i = 0; i < numsSize - 1; i++) {
        for (int j = i + 1; j < numsSize; j++) {
            if (nums[i] + nums[j] == target) {
                int* result = (int*)malloc(2 * sizeof(int));
                result[0] = i;
                result[1] = j;
                *returnSize = 2;
                return result;
            }
        }
    }
    return NULL;
}
```

Solution in Java:

```
/*
  * Brute Force Method.
*/
class Solution {
    public int[] twoSum(int[] nums, int target) {
        int[] retArr = new int[2];
        for(int i = 0; i < nums.length - 1; i++) {
            for(int j = i + 1; j < nums.length; j++) {
```

```
                if(nums[i] + nums[j] == target) {
                    retArr[0] = i;
                    retArr[1] = j;
                }
            }
        }
        return retArr;
    }
}
```

Explanation: In this solution, we have a function `twoSum` that takes an array of integers `nums`, its size `numsSize`, the target value `target`, and a pointer `returnSize` to store the size of the result array. The function uses nested loops to iterate over all possible pairs of elements in the array and checks if their sum equals the target. If a solution is found, it dynamically allocates memory for the result array, stores the indices of the two numbers, sets `returnSize` to 2, and returns the result. If no solution is found, it returns `NULL`.