#include <iostream>

#include <unordered\_map>

#include <vector>

std::vector<int> twoSum(std::vector<int>& nums, int target) {

std::unordered\_map<int, int> numMap;

std::vector<int> result;

for (int i = 0; i < nums.size(); ++i) {

int complement = target - nums[i];

if (numMap.find(complement) != numMap.end()) {

result.push\_back(numMap[complement]);

result.push\_back(i);

break;

}

numMap[nums[i]] = i;

}

return result;

}

int main() {

// Example 1

std::vector<int> nums1 = {2, 7, 11, 15};

int target1 = 9;

std::vector<int> result1 = twoSum(nums1, target1);

std::cout << "Example 1 Output: [" << result1[0] << ", " << result1[1] << "]" << std::endl;

// Example 2

std::vector<int> nums2 = {3, 2, 4};

int target2 = 6;

std::vector<int> result2 = twoSum(nums2, target2);

std::cout << "Example 2 Output: [" << result2[0] << ", " << result2[1] << "]" << std::endl;

// Example 3

std::vector<int> nums3 = {3, 3};

int target3 = 6;

std::vector<int> result3 = twoSum(nums3, target3);

std::cout << "Example 3 Output: [" << result3[0] << ", " << result3[1] << "]" << std::endl;

return 0;

}