**Problem Name: Count pairs with given sum**

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**:

**Input Format:**

First line of the input contain integer n (size of list)

Second line contain n space separated integer list values.

Last line contain integer value pos representing value of node to delete.

Ex:

5

1 2 3 4 5

1

**Output Format:** Print linked list after removing node having value pos

**Constraints:**

**Examples:**

**Brute force Solution:**

# Explanation: A **simple solution** is to traverse each element and check if there’s another number in the array which can be added to it to give sum.

**Code:**

**Time Complexity**: O(N2)

**Space Complexity:** O(1)

**Optimized Solution:**

# Explanation:

1. Create a map to store frequency of each number in the array. (Single traversal is required)
2. In the next traversal, for every element check if it can be combined with any other element (other than itself!) to give the desired sum. Increment the counter accordingly.
3. After completion of second traversal, we’d have twice the required value stored in counter because every pair is counted two times. Hence divide count by 2 and return.

**Code:**

**Time Complexity**: O(n)

**Space Complexity:** O(n)