**Problem Name:** Valid triangle number

**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:** Just traverse the array with all possibility of satisfying initial condition, if yes increase the count value.

**Code:**

**Time Complexity**: O(n^3)

**Space Complexity:** O(1)

**Optimized Solution:**

**Explanation:**

Let us assume there are 3 sides of a triangle named a ,b ,c  
and we are going to choose a,b,c in such an order, so that **a<b<c**  
Now we know condition to make a traingle is  
(1). **a+b>c**  
(2). **b+c>a**  
(3). **c+a>b**  
Since a<b<c , the inequalities (2) and (3) are always satisfied irrespective of their values (as long as a<b<c is satisfied). So we need not check them and **All we need to check is inequality (1)**  
We first sort the vector/array using STL for our simplicity  
So we traverse the given vector/array backwards (from larger to smaller value)  
The first for loop is for side c and the second for loop is for side b (placed before side c in the sorted array)  
Now we need some side a such that a+b>c or **a>c-b**  
And we use upper\_bound function to find the occurence of first such valid side a  
And we simply add all such side a elements to our answer(Note that in this case even duplicate values get handled very well)

**Code:**

**Time Complexity**: O(n^2)

**Space Complexity:** O(1)