**Problem Name:** K diff pairs in an array

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**:

**Input Format:**

**Output Format:**

**Constraints:**

**Examples:**

**Brute force Solution:**

**Explanation:**

#### **2. USING TWO POINTERS:**

We are aware of the fact that for a pair to be counted as an answer, **both the elements (x and x+k), need to be in the array.**  
In this approach, **first we sort the array** and maintain 2 pointers.

* 1st Pointer --> 1st Element of the Pair
* 2nd Pointer --> 2nd Element of the Pair

We set the 1st pointer at the 0th index and 2nd pointer at the 1st index. Then ,

1. Move the 2nd pointer until **2nd pointer - 1st pointer >=k**.
2. If the **2nd pointer - 1st pointer ==k**, then **increment the answer and move the 2nd pointer to the next greater element.**
3. Move the **1st pointer to the next greater element**.

Follow the above procedure until **the 2nd pointer reaches the end of the array**.

**Code:**

**Time Complexity**: O(NlogN)

**Space Complexity: O(1)**

**Optimized Solution:**

Explanation:

#### **1. USING MAPS :**

We are aware of the fact that for a pair to be counted as an answer , **both the elements ( x and x+k ) , need to be in the array**.  
So we simply **create a map and store the frequency** of each element in the map.  
Now we traverse the map and for **each element 'x'** , we **check if 'x+k' exists in the map** . If it does , then it means **a unique pair can be formed** and hence, we **increment the answer**.

##### **EDGE CASE :**

The only edge case is the situation where**k=0**. If k=0 , instead of finding 'x+k' , we **check if the frequency of 'x'>1**. If it is , then we**increment the answer** .  
Else , we don't **increment the answer , as the frequency of x=1 and hence it can't form a pair with itself**.

**Code:**

**Time Complexity**: O(N)

**Space Complexity:** O(N)