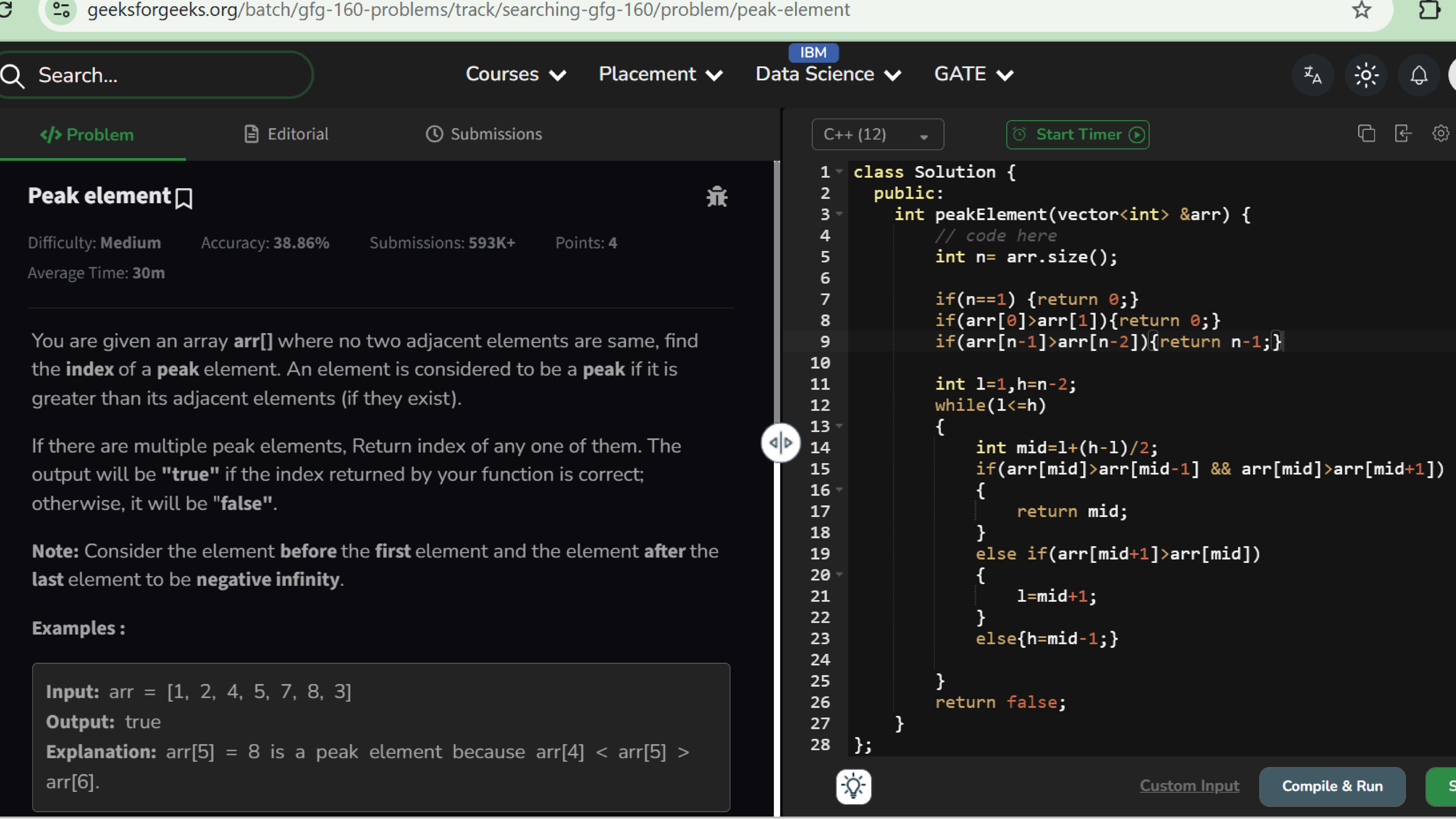


Peak element

Difficulty: **Medium** Accuracy: **38.86%** Submissions: **593K+** Points: **4**

Average Time: **30m**

You are given an array **arr[]** where no two adjacent elements are same, find the **index** of a **peak** element. An element is considered to be a **peak** if it is greater than its adjacent elements (if they exist).

If there are multiple peak elements, Return index of any one of them. The output will be **"true"** if the index returned by your function is correct; otherwise, it will be "**false"**.

Note: Consider the element **before** the **first** element and the element **after** the **last** element to be **negative infinity**.

Examples :

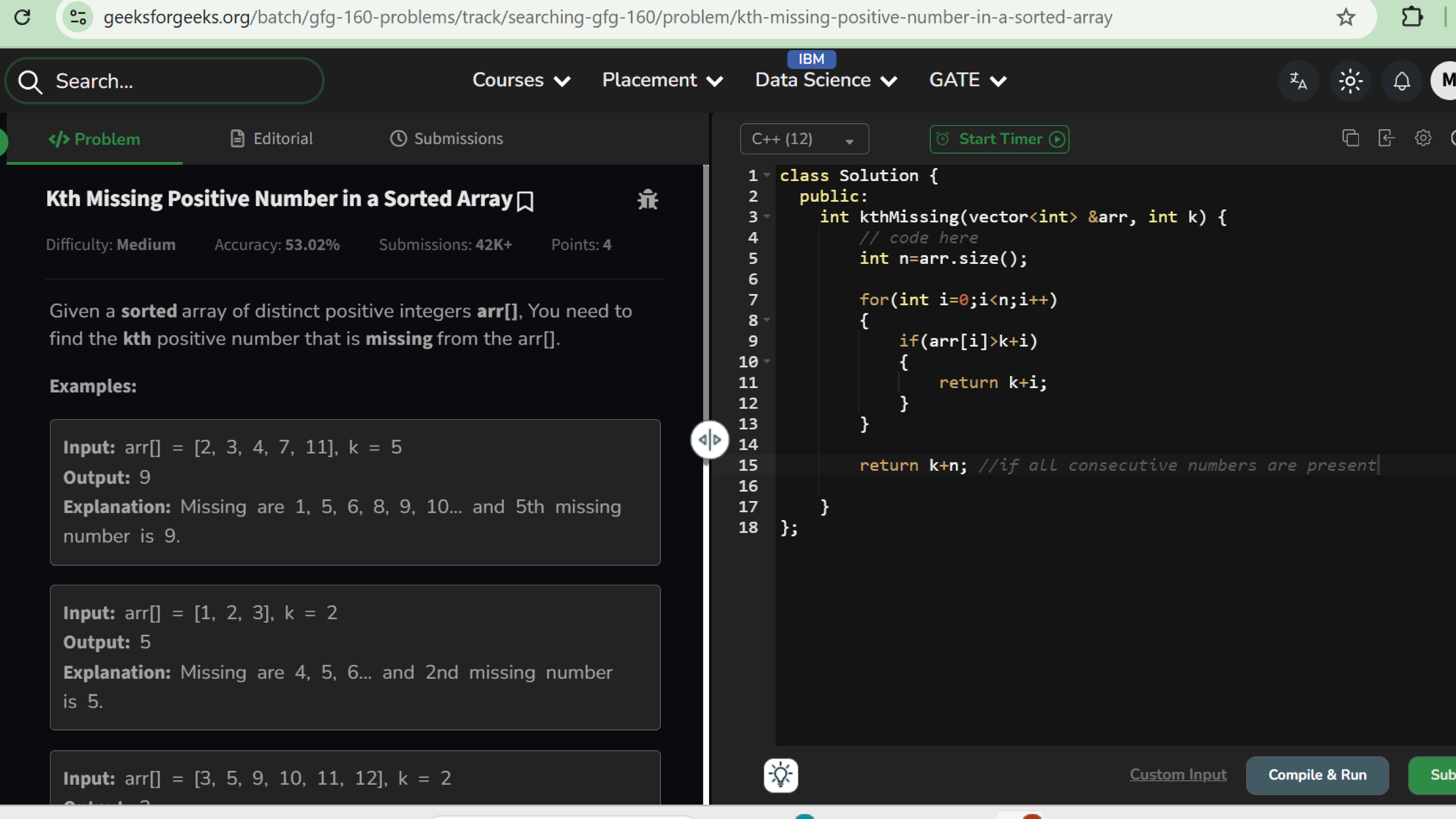
Input: arr = [1, 2, 4, 5, 7, 8, 3]
Output: true
Explanation: arr[5] = 8 is a peak element because arr[4] < arr[5] > arr[6].

```cpp
class Solution {
  public:
    int peakElement(vector<int> &arr) {
        // code here
        int n= arr.size();

        if(n==1) {return 0;}
        if(arr[0]>arr[1]){return 0;}
        if(arr[n-1]>arr[n-2]){return n-1;}

        int l=1,h=n-2;
        while(l<=h)
        {
            int mid=l+(h-l)/2;
            if(arr[mid]>arr[mid-1] && arr[mid]>arr[mid+1])
            {
                return mid;
            }
            else if(arr[mid+1]>arr[mid])
            {
                l=mid+1;
            }
            else{h=mid-1;}

        }
        return false;
    }
};
```


geeksforgeeks.org/batch/gfg-160-problems/track/searching-gfg-160/problem/kth-missing-positive-number-in-a-sorted-array
Search...
Courses
Placement
IBM
Data Science
GATE
Problem
Editorial
Submissions
C++ (12)
Start Timer
Custom Input
Compile & Run
Sub

Kth Missing Positive Number in a Sorted Array

Difficulty: **Medium** Accuracy: **53.02%** Submissions: **42K+** Points: **4**

Given a **sorted** array of distinct positive integers **arr[]**. You need to find the **kth** positive number that is **missing** from the arr[].

Examples:

Input: arr[] = [2, 3, 4, 7, 11], k = 5
Output: 9
Explanation: Missing are 1, 5, 6, 8, 9, 10... and 5th missing number is 9.

Input: arr[] = [1, 2, 3], k = 2
Output: 5
Explanation: Missing are 4, 5, 6... and 2nd missing number is 5.

Input: arr[] = [3, 5, 9, 10, 11, 12], k = 2

```cpp
class Solution {
  public:
    int kthMissing(vector<int> &arr, int k) {
        // code here
        int n=arr.size();

        for(int i=0;i<n;i++)
        {
            if(arr[i]>k+i)
            {
                return k+i;
            }
        }

        return k+n; //if all consecutive numbers are present

    }
};
```