2657. Find the Prefix Common Array of Two Arrays

You are given two **0-indexed** integer permutations `A` and `B` of length `n`.

A **prefix common array** of `A` and `B` is an array `C` such that `C[i]` is equal to the count of numbers that are present at or before the index `i` in both `A` and `B`.

Return *the* ***prefix common array*** *of* `A` *and* `B`.

A sequence of `n` integers is called a **permutation** if it contains all integers from `1` to `n` exactly once.

Example 1:

```
Input: A = [1,3,2,4], B = [3,1,2,4]
Output: [0,2,3,4]
Explanation: At i = 0: no number is common, so C[0] = 0.
At i = 1: 1 and 3 are common in A and B, so C[1] = 2.
At i = 2: 1, 2, and 3 are common in A and B, so C[2] = 3.
At i = 3: 1, 2, 3, and 4 are common in A and B, so C[3] = 4.
```

Example 2:

```
Input: A = [2,3,1], B = [3,1,2]
Output: [0,1,3]
Explanation: At i = 0: no number is common, so C[0] = 0.
At i = 1: only 3 is common in A and B, so C[1] = 1.
At i = 2: 1, 2, and 3 are common in A and B, so C[2] = 3.
```

Constraints:

- `1 <= A.length == B.length == n <= 50`
- `1 <= A[i], B[i] <= n`
- `It is guaranteed that A and B are both a permutation of n integers.`

2657. Find the Prefix Common Array of Two Arrays

Overview

We are given two arrays, `A` and `B`, each containing a shuffled list of numbers from `1` to `n`. Our task is to construct a new array `C`, where each element `C[i]` represents the count of numbers from `1` to `i + 1` that are present in both `A` and `B` up to that index.

For example, consider `A = [1, 3, 2, 4]` and `B = [3, 1, 2, 4]`:

- At `i = 0` (first element): No numbers are common between `A` and `B` yet, so `C[0] = 0`.
- At `i = 1`: The numbers `1` and `3` are common in both arrays, so `C[1] = 2`.
- At `i = 2`: The numbers `1`, `2`, and `3` are common, so `C[2] = 3`.
- At `i = 3`: All four numbers, `1`, `2`, `3`, and `4`, are common in both arrays, so `C[3] = 4`.

Thus, the resulting array is `C = [0, 2, 3, 4]`.

2657. Find the Prefix Common Array of Two Arrays

```
/*
   Counting: track encountered elements so far
   Time compelxity: O(n)
   Space complexity: O(n)
*/
class Solution {
   public:
      std::vector<int> findThePrefixCommonArray(std::vector<int>& A, std::vector<int>& B){
         int n=A.size();
         std::vector<int> seen(n+1,0);
         std::vector<int> ans(n);
         ans[0]=int(A[0]==B[0]);
         seen[A[0]]=seen[B[0]]=1;
         for(int i=1;i<n;++i){
            ans[i]=ans[i-1]+seen[A[i]]+seen[B[i]]+(A[i]==B[i]);
            seen[A[i]]=1;
            seen[B[i]]=1;
         }
         return ans;
      }
};
```