|  |  |
| --- | --- |
| **Good Integers duplicate in C++** | |
| #include <iostream>  #include <vector>  #include <algorithm>  using namespace std;  int GoodIntegers(int arr[], int n) {  sort(arr, arr + n); // Sort the array  int ans = 0;  int lessCount = 0;  if (arr[0] == 0) {  ans++;  }  for (int i = 1; i < n; ++i) {  if (arr[i] != arr[i - 1]) {  lessCount = i;  }  if (arr[i] == lessCount) {  ans++;  }  }  return ans;  }  int main() {  int arr[] = {0, 1, 5, 7, 8, 9, 4};  int n = sizeof(arr) / sizeof(arr[0]);  cout << GoodIntegers(arr, n) << endl;  return 0;  } | **Goal of the Function:**  Count how many elements in the array are **equal to the number of elements less than it**.  **🔄 Step-by-step Dry Run**  **➤ Step 1: Sort the array**  Initial array: {0, 1, 5, 7, 8, 9, 4} Sorted array: {0, 1, 4, 5, 7, 8, 9} n = 7 Variables: ans = 0, lessCount = 0   | **Index (i)** | **arr[i]** | **arr[i-1]** | **arr[i] != arr[i-1]** | **lessCount** | **arr[i] == lessCount** | **ans** | | --- | --- | --- | --- | --- | --- | --- | | 0 | 0 | - | - | 0 | ✅ (0 == 0) | 1 | | 1 | 1 | 0 | ✅ | 1 | ✅ (1 == 1) | 2 | | 2 | 4 | 1 | ✅ | 2 | ❌ (4 != 2) | 2 | | 3 | 5 | 4 | ✅ | 3 | ❌ (5 != 3) | 2 | | 4 | 7 | 5 | ✅ | 4 | ❌ (7 != 4) | 2 | | 5 | 8 | 7 | ✅ | 5 | ❌ (8 != 5) | 2 | | 6 | 9 | 8 | ✅ | 6 | ❌ (9 != 6) | 2 |   **✅ Final Answer: 2**  The two good integers are:   * 0: there are 0 elements less than it → good * 1: there is 1 element less than it → good |
| 2 | |