|  |  |
| --- | --- |
| **Find Transition in C++** | |
| #include <iostream>  #include <vector>  using namespace std;  int findTransition(vector<int>& arr) {  int tp = -1;  int lo = 0;  int hi = arr.size() - 1;  while (lo <= hi) {  int mid = lo + (hi - lo) / 2;  if (arr[mid] == 1) {  tp = mid;  hi = mid - 1; // Look for earlier occurrences on the left side  } else {  lo = mid + 1; // If arr[mid] is 0, move to the right half  }  }  return tp;  }  int main() {  // Hardcoded input  vector<int> arr = {0, 0, 0, 0, 1, 1};  // Call the findTransition function to find the index of the first occurrence of 1  int ans = findTransition(arr);  // Print the index of the first occurrence of 1  cout << ans << endl;  return 0;  } | **Input:**  arr = {0, 0, 0, 0, 1, 1}  **🎯 Goal:**  Find the **index of the first occurrence of 1** using **binary search**.  **🔍 Dry Run Table:**   | **Iteration** | **lo** | **hi** | **mid** | **arr[mid]** | **tp** | **Action Taken** | | --- | --- | --- | --- | --- | --- | --- | | 1 | 0 | 5 | 2 | 0 | -1 | Move right: lo = mid + 1 = 3 | | 2 | 3 | 5 | 4 | 1 | 4 | Move left: hi = mid - 1 = 3 | | 3 | 3 | 3 | 3 | 0 | 4 | Move right: lo = mid + 1 = 4 |   **✅ Final Values:**   * tp = 4 * So, the **first occurrence of 1 is at index 4**.   🖨️ Output:  4 |
| 4 | |