Strivers-A2Z-DSA-Sheet-main\02.Binary Search\1D Arrays\01.Find_x_in_sorted_array.cpp

```cpp
/*
QUESTION:-
Given an array of integers nums which is sorted in ascending order, and an integer target, write a function to search target in nums. If target exists, then return its index. Otherwise, return -1.

Example 1:
Input: nums = [-1,0,3,5,9,12], target = 9
Output: 4
Explanation: The target value 9 exists in the nums array, and its index is 4.

Example 2:
Input: nums = [-1,0,3,5,9,12], target = 2
Output: -1
Explanation: The target value 2 does not exist in the nums array, so return -1.
*/

/*
APPROACH:-
1. Initialize low as 0 and high as the last index of the array.
2. Iterate using a while loop until low is less than or equal to high.
3. Calculate the middle index using the formula mid = low + (high - low) / 2.
4. Compare the target value with the element at the middle index:
   - If they are equal, return the middle index.
   - If the target is less than the element, update high to mid - 1 and continue the search in the left half.
   - If the target is greater than the element, update low to mid + 1 and continue the search in the right half.
5. If the target is not found, return -1.
*/

//CODE:-
int search(vector<int>& nums, int target) {
    int low = 0, high = nums.size() - 1;
    while (low <= high) {
        int mid = low + (high - low) / 2;
        if (nums[mid] == target)
            return mid;
        else if (nums[mid] > target)
            high = mid - 1;
        else
            low = mid + 1;
    }
    return -1;
}


// TIME COMPLEXITY: O(log n)
// - The algorithm divides the search space in half at each step, resulting in a logarithmic time complexity.

// SPACE COMPLEXITY: O(1)
```