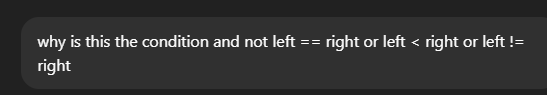
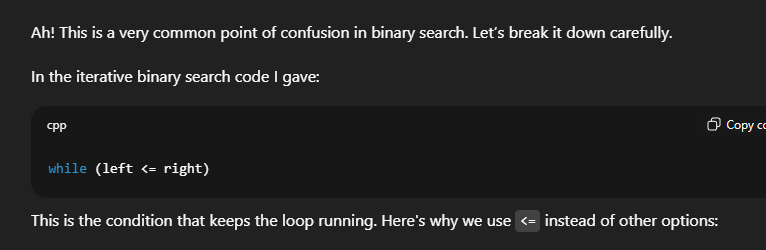
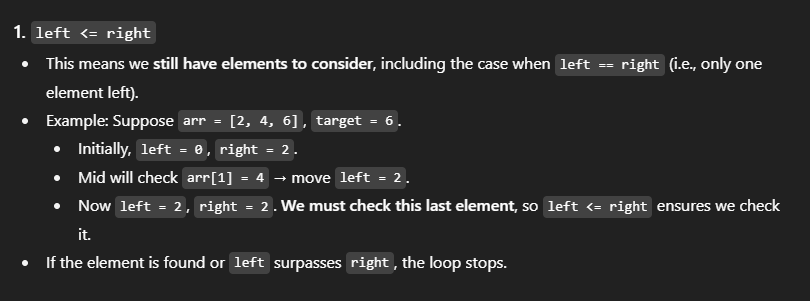
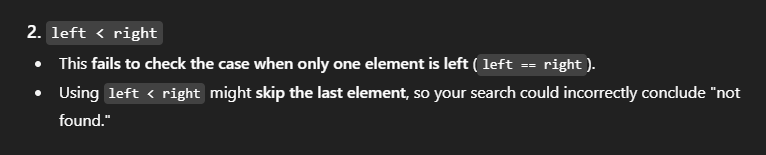
While loop condition reason 🡪

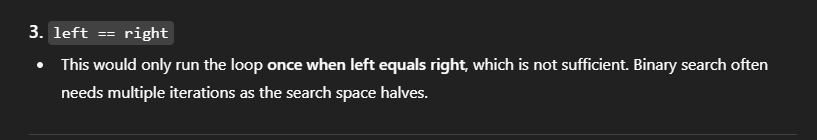


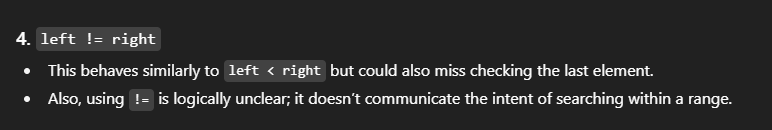




When left == right, 1 element remains. There is a possibility that the last element may be the key





  
original code 🡪

// Binary Search

#include<iostream>

using namespace std;

// Function to take input for the array

void arrayInput(int arr[], int size) {

for (int i = 0; i < size; i++) {

cout << "Enter element " << i + 1 << ": ";

cin >> arr[i];

}

}

int BinarySearch(int arr[], int Size, int Key) {

int start = 0;

int end = Size - 1;

while (start <= end) {

int mid = (start + end) / 2;

if (arr[mid] == Key) {

return mid;

}

// for readability though the return function already enforces mutual exclusivity

else if (arr[mid] > Key) {

// key is smaller than the middle element, shift end backwards

end = mid - 1;

}

else {

// key is bigger than the middle element, shifting start forward

start = mid + 1;

}

}

return -1;

}

int main() {

int arr[1000]; // Maximum array size

int size = 0;

int key = -1;

// Input array size

cout << "Enter size of array (max 1000): ";

cin >> size;

// Take array elements as input

arrayInput(arr, size);

// Input element to search

cout << "\nEnter element to find: ";

cin >> key;

cout<<"Element found at Position (-1 for not found) : " << BinarySearch(arr, size, key);

return 0;

}

