1. Insert Element

Difficulty: *Easy*

Question:

Given a sorted integer array *'arr'* containing unique elements, and an element *'n'*, determine and display the two consecutive elements between which *'n'* can be inserted while preserving the sorted order of the array.

Constraint: *Time Complexity : O(log N)*

Example 1:

Input: arr= [1, 5, 7, 8, 10, 19], n = 6

Output: 5 and 7

Explanation: Element 6 should be inserted between 5 and 7.

Example 2:

Input: arr= [-1, 2, 7, 9], n = -2

Output: NULL and -1

Explanation: Element -2 should be inserted before -1. Hence the previous value is NULL.

Example 3:

Input: arr= [-1, 2, 7, 9], n = 10

Output: 9 and NULL

Explanation: Element 10 should be inserted after 9. Hence the next value is NULL.

Example 3:

Input: arr= [5, 15, 20, 25, 40, 60, 100], n = 25

Output: 25 is already present in the array.

(Code given below! Try on your own before viewing the code)

Solution : (C++)

```cpp
#include<bits/stdc++.h>
using namespace std;
int binarySearch(int arr[], int start, int end, int n){
    int ans = -1;

    while(start <= end){
        int mid = (start + end) / 2;

        if(arr[mid] <= n){
            ans = mid;
            start = mid + 1;
        }
        else end = mid - 1;
    }
    return ans;
}

int main(){
    int arr[] = {1, 5, 7, 8, 10, 19};
    int n = 6;
    int len = sizeof(arr)/sizeof(arr[0]);

    // Using Binary Search, locate the index where to be inserted
    int idx = binarySearch(arr, 0, len - 1, n);

    if(idx == len - 1 && arr[idx] != n)
        cout << endl << arr[idx] << " and NULL \n";

    else if(idx == -1)
        cout << endl << "NULL and " << arr[0] << endl;

    else if(arr[idx] == n)
        cout << "\n Element " << n << " is already present! \n";

    else
        cout << endl << arr[idx] << " and " << arr[idx + 1] << endl;

    return 0;
}
```