Started on	Tuesday, 8 October 2024, 1:51 PM
State	Finished
Completed on	Tuesday, 8 October 2024, 2:10 PM
Time taken	19 mins 19 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Problem Statement:

Given a sorted array of integers say arr[] and a number x. Write a recursive program using divide and conquer strategy to check if there exist two elements in the array whose sum = x. If there exist such two elements then return the numbers, otherwise print as "No".

Note: Write a Divide and Conquer Solution

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Sum Value

Output Format

First Line Contains Integer – Element1

Second Line Contains Integer – Element2 (Element 1 and Elements 2 together sums to value "x")

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int binarySearch(int arr[], int low, int high, int target) {
    if (high >= low) {
        int mid = low + (high - low) / 2;
        if (arr[mid] == target)
            return mid;
        if (arr[mid] > target)
            return binarySearch(arr, low, mid - 1, target);
        return binarySearch(arr, mid + 1, high, target);
    }
    return -1;
}
void findTwoElements(int arr[], int n, int x) {
    for (int i = 0; i < n - 1; i++) {
        int complement = x - arr[i];
        int idx = binarySearch(arr, i + 1, n - 1, complement);
        if (idx != -1) {
            printf("%d\n%d\n", arr[i], arr[idx]);
            return;
        }
    }
    printf("No\n");
}
int main() {
    int n, x;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    scanf("%d", &x);
    findTwoElements(arr, n, x);
    return 0;
}
```

	Input	Expected	Got	
✔	4 2 4 8 10 14	4 10	4 10	✔

	Input	Expected	Got	
✔	5 2 4 6 8 10 100	No	No	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 3-Finding Floor Value

Jump to...

5-Implementation of Quick Sort ►