<u>Dashboard</u> / <u>My courses</u> / <u>CS23331-DAA-2023-CSE</u> / <u>Divide and Conquer</u> / <u>4-Two Elements sum to x</u>

Started on	Tuesday, 8 October 2024, 2:21 PM
State	Finished
Completed on	Tuesday, 8 October 2024, 8:43 PM
Time taken	6 hours 21 mins
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

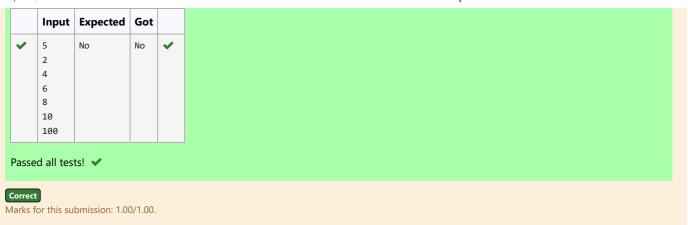
First Line Contains Integer - Element1

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

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
 2 int find_pair(int arr[], int left, int right, int x, int *num1, int *num2) {
 3 1
        if (left >= right) {
 4
            return 0;
 5
 6
 7
        int sum = arr[left] + arr[right];
        if (sum == x) {
 8
 9
            *num1 = arr[left];
            *num2 = arr[right];
10
11
            return 1;
12
        }
13
        else if (sum < x) {</pre>
14
15
            return find_pair(arr, left + 1, right, x, num1, num2);
16
        }
17
        else {
            return find_pair(arr, left, right - 1, x, num1, num2);
18
19
        }
20
21 v int main() {
22
        int n, x;
        scanf("%d", &n);
23
24
        int arr[n];
        for (int i = 0; i < n; i++) {
25
26
            scanf("%d", &arr[i]);
27
        }
        scanf("%d", &x);
28
29
        int num1, num2;
30
        if (find_pair(arr, 0, n - 1, x, &num1, &num2)) {
31
            printf("%d\n%d\n", num1, num2);
32
        } else {
            printf("No\n");
33
34
35
36
        return 0;
37 }
```

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



■ 3-Finding Floor Value

Jump to... \$

5-Implementation of Quick Sort ►