<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	Friday, 4 October 2024, 1:54 PM
State	Finished
Completed on	Friday, 4 October 2024, 1:54 PM
Time taken	11 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

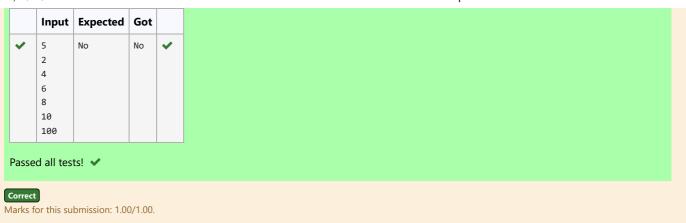
First Line Contains Integer - Element1

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

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
 2 v int findPairRecursive(int arr[], int low, int high, int x) {
 3 1
        if (low >= high) {
 4
             return 0;
 5
 6
        int left = low;
 7
        int right = high;
        while (left < right) {</pre>
 8
 9
             int sum = arr[left] + arr[right];
10
             if (sum == x) {
                 printf("%d\n", arr[left]);
11
12
                 printf("%d\n", arr[right]);
13
                 return 1;
             } else if (sum < x) {
14
15
                 left++;
16
             } else {
17
                 right--;
18
19
20
        return 0;
21
22
   void checkPair(int arr[], int n, int x) {
23
        if (!findPairRecursive(arr, 0, n - 1, x)) {
             printf("No\n");
24
25
        }
26
27 v int main() {
        int n, x;
scanf("%d", &n);
28
29
30
        int arr[n];
31
        for (int i = 0; i < n; i++) {</pre>
32
             scanf("%d", &arr[i]);
33
        scanf("%d", &x);
34
        checkPair(arr, n, x);
35
36
         return 0;
37
    }
38
```

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



■ 3-Finding Floor Value

5-Implementation of Quick Sort ►