4.d. Two Elements Sum to X

Aim: 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")

Algorithm:

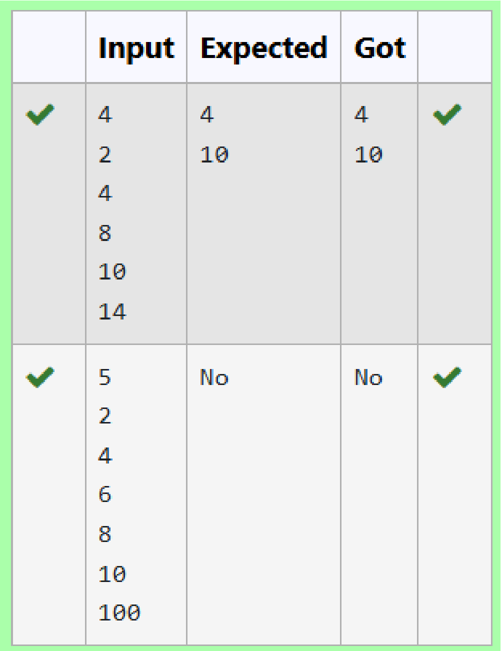
```
int findPairWithSum(arr, left, right, x){

  // Base case: if there are no more pairs to check

  if left >= right

     print "No" // No pair found

     return


  // Calculate the sum of the elements at the left and right indices

  sum = arr[left] + arr[right]


  // Check if the sum is equal to x

  if sum is equal to x

     print arr[left] // Print the first element of the pair print

     arr[right] // Print the second element of the pair return


  // If the sum is less than x, move the left index up
```

```
    if sum < x
        findPairWithSum(arr, left + 1, right, x) // Recursive call with increased left index else
        findPairWithSum(arr, left, right - 1, x) // Recursive call with decreased right index
}
function main()
    initialize n // number of elements in the array
    read n from user

    initialize arr of size n // array to hold input values

    // Read values into the array
    for i from 0 to n - 1
        read arr[i] from user

    initialize x // the target sum value
    read x from user

    // Call the findPairWithSum function
    findPairWithSum(arr, 0, n - 1, x)
```

Program:

```
#include <stdio.h>

void findPairWithSum(int arr[], int left, int right, int x) { if
    (left >= right) {
        //No pair found
        printf("No\n");
        return;
    }

    int sum = arr[left] + arr[right];

    if (sum == x){
        // If the pair is found
        printf("%d\n%d\n", arr[left], arr[right]);
        return;
    }

    if (sum < x){

        findPairWithSum(arr, left + 1, right, x);
    }
    else{

        findPairWithSum(arr, left, right - 1, x);
    }
}
int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int x;
    scanf("%d", &x);
    findPairWithSum(arr, 0, n - 1, x);
}
```

Output:



	Input	Expected	Got	
✓	4 2 4 8 10 14	4 10	4 10	✓
✓	5 2 4 6 8 10 100	No	No	✓