

Aim:

Write a program to **search** the given element from a list of elements with **binary search** technique using **recursion**.

At the time of execution, the program should print the message on the console as:

Enter value of n :

For example, if the user gives the **input** as:

Enter value of n : 5

Next, the program should print the following messages one by one on the console as:

Enter 5 elements :

if the user gives the **input** as:

Enter 5 elements : 33 55 22 44 11

then the program should **print** the result as:

After sorting the elements are : 11 22 33 44 55

Next, the program should print the message on the console as:

Enter key element :

if the user gives the **input** as:

Enter key element : 11

then the program should **print** the result as:

The given key element 11 is found at position : 0

Similarly, if the key element is given as **18** for the above example then the program should print the output as:

The given key element 18 is not found

Note: Write the functions **read()**, **bubbleSort()**, **display()** and **binarySearch()** in **BinarySearch.c**

Source Code:

BinarySearch.c

```
#include<stdio.h>
void main()
{
    int a[20],n,key,flag;
    printf("Enter value of n : ");
    scanf("%d", &n);
    read(a, n);
```

```
bubbleSort(a, n);
printf("After sorting the elements are : ");
display(a, n);
printf("Enter key element : ");
scanf("%d", &key);
flag = binarySearch(a, 0, n - 1, key);
if(flag == -1)
{
    printf("The given key element %d is not found\n", key);
}
else
{
    printf("The given key element %d is found at position : %d\n", key, flag);
}
}
void read(int a[], int n)
{
    int i;
    printf("Enter %d elements : ",n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
}
void display(int a[], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d ",a[i]);
    }
    printf("\n");
}
void bubbleSort(int a[], int n)
{
    int i,j,temp;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n-1;j++)
        {
            if(a[j+1]<a[j])
            {
                temp=a[j+1];
                a[j+1]=a[j];
                a[j]=temp;
            }
        }
    }
}
int binarySearch(int a[],int x,int n,int key)
{
    int low=0,high=n-1,mid;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(a[mid]==key)
```

```

    {
        x=mid;
        return x;
        break;
    }
    else if(a[mid]<key)
    {
        low=mid+1;
    }
    else
    {
        high=mid-1;
    }
}
return -1;
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter value of n : 5
Enter 5 elements : 33 55 22 44 11
After sorting the elements are : 11 22 33 44 55 11
Enter key element : 11
The given key element 11 is found at position : 0

Test Case - 2
User Output
Enter value of n : 4
Enter 4 elements : 23 9 45 18
After sorting the elements are : 9 18 23 45 24
Enter key element : 24
The given key element 24 is not found