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 Program912a.c

Source Code:

Program912.c

```c
#include <stdio.h>
#include "Program912a.c"
void main() {
    int a[20], n, key, flag;
    printf("Enter value of n : ");
    scanf("%d", &n);
    read(a, n);
    bubbleSort(a, n);
```

```c
      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);
      }
}
```

Program912a.c

```c
void read(int a[],int n)
{
   int i;
   printf("Enter %d elements : ",n);
   for(i=0;i<n;i++)
   scanf("%d",&a[i]);
}
void bubbleSort(int a[],int n)
{
   int i,j,temp;
   for(i=0;i<n-1;i++)
   {
      for(j=0;j<n-1;j++)
      {
         if(a[j]>a[j+1])
         {
            temp=a[j];
            a[j]=a[j+1];
            a[j+1]=temp;
         }
      }
   }
}
void display(int a[],int n)
{
   int i;
   for(i=0;i<n;i++)
   printf("%d ",a[i]);
   printf("\n");
}
int binarySearch(int a[],int l,int h,int key)
{
   int mid,flag;
   if(l>h)
   return -1;
   else
   {
      mid=(l+h)/2;
      if(a[mid]==key)
      {
          flag=mid;
```

```
            return flag;
        }
        else
        {
            if(a[mid]<key)
            binarySearch(a,mid+1,h,key);
            else
            binarySearch(a,l,mid-1,key);
        }
    }
}
```

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 67 45 18
After sorting the elements are : 18 23 45 67 24
Enter key element : 24
The given key element 24 is not found

Test Case - 3
User Output
Enter value of n : 6
Enter 6 elements : 10 20 18 9 11 15
After sorting the elements are : 9 10 11 15 18 20 18
Enter key element : 18
The given key element 18 is found at position : 4