```
//Created By Ritwik Chandra Pandey on 8th Nov 2021
//Binary Search(Recursion)



#include <stdio.h>

// Recursive implementation of the binary search algorithm to return
// the position of `target` in subarray nums[low…high]


int binarySearch(int nums[], int low, int high, int target)
{
// Base condition (search space is exhausted)
if (low > high) {
return -1;
}

// find the mid-value in the search space and
// compares it with the target

int mid = (low + high)/2;// overflow can happen
// int mid = low + (high - low)/2;

// Base condition (target value is found)
if (target == nums[mid]) {
return mid;
}

// discard all elements in the right search space,
// including the middle element
else if (target < nums[mid]) {
return binarySearch(nums, low, mid - 1, target);
}

// discard all elements in the left search space,
// including the middle element
else {
return binarySearch(nums, mid + 1, high, target);
}
```

```
}

int main() {
int a[20], i,j, n, key,low, high,temp;
printf("Enter value of n : ");
    scanf("%d", &n);
for(i=0;i<n;i++)
 { printf("Enter element for a[%d] : ",i);
     scanf("%d",&a[i]);

 }
printf("Enter key element : ");
 scanf("%d", &key);
  for(i=0;i<n-1;i++){
       for(j=0;j<n-i-1;j++){
       }
     }

if(a[j]>a[j+1]){
temp=a[j];
 a[j] = a[j+1];
   a[j+1] = temp; }

printf("After sorting the elements in the array are\n");

 for(i=0;i<n;i++){
     printf("Value of a[%d] = %d\n",i,a[i]); }

low = 0; high = n-1;


int index = binarySearch(a, low, high, key);

if (index != -1) {
printf("Element found at index %d", index);
}
else {
printf("Element not found in the array");
}
return 0;
```

```
}
```