

```
1 //日期: 2018/ 时间:
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 //A[]为严格递增序列, left为二分下界, right为二分上界, x为欲查询的数
6 //二分区间为左闭右闭[left,right], 传入的初值为[0,n-1]
7 int binarySearch(int A[],int left,int right,int x){
8     int mid;
9     while(left <= right){           //如果left>right就没办法形成闭区间了
10         mid = (right + left) / 2;    //取中点
11         if(A[mid] == x) return mid; //找到x,返回下标
12         else if(A[mid] > x) {        //中间的数大于x
13             right = mid - 1;         //往左子区间[left,mid-1]查找
14         }else{                      //中间的数小于x
15             left = mid + 1;          //往右子区间[mid+1,right]查找
16         }
17     }
18     return -1;                      //查找失败, 返回-1
19 }
20
21 //如果递增序列A中的元素可能重复, 求出序列中第一个大于等于x的位置L, 以及第一个大于x的 ↗
    位置R,
22 //这样元素x在序列中的存在区间就是左闭右开区间[L,R)
23
24 //1、求序列中第一个大于等于x的元素的位置
25 //二分的初试区间应该能覆盖到所有可能返回的结果, 二分下界是0, 二分上界是n而不是n-1, 因 ↗
    为欲查元素可能比所有的元素都大
26 //二分上下界为左闭右闭区间[left,right],传入初值为[0,n]
27 int lower_bound(int A[],int left,int right,int x){
28     int mid;
29     while(left < right){            //对[left,right]来说, left==right意味着找到唯一的 ↗
        位置
30         mid = (left+right)/2;        //取中点
31         if(A[mid] >= x){
32             right = mid;             //往左子区间查找
33         }else{
34             left = mid+1;
35         }
36     }
37     return left;
38 }
39
40 //2、求序列中第一个大于x的元素的位置
41 //二分上下界为左闭右闭区间[left,right],传入初值为[0,n]
42 int upper_bound(int A[],int left,int right,int x){
43     int mid;
44     while(left < right){
45         mid = (left + right)/2;
46         if(A[mid] > x){
47             right = mid;
48         }else{                      //A[mid] <= x
49             left = mid+1;
50         }
51     }
52     return left;
53 }
```

```
54
55 int main(){
56     const int n=10;
57     int A[n] = {1,3,4,6,7,8,10,11,12,15};
58
59     printf("%d %d\n",binarySearch(A,0,n-1,6),binarySearch(A,0,n-1,9));
60
61     return 0;
62 }
63
64
65
66
```