

12 数组应用1查找

主教练：党东



【例题】查找

【问题描述】 在 n 个整数查找是否出现某一个数，若找到则输出在数组中的位置，否则输出“Fail!”。

【输入格式】 第一行为一个整数 n ($n \leq 100$)，第二行为 n 个用空格分开的整数，第三行为要查找的数字 x 。

【输出格式】 在 n 个数字中 x 出现则输出在数组中的位置，否则输出“Fail!”

【样例输入】

5

5 6 7 8 9

7

【样例输出】 3

题目分析

```
k=0; //先假设找不到这个数字
cin>>n;
for(i=1;i<=n;i++)cin>>a[i];
cin>>x;
for(i=1;i<=n;i++)
    if(a[i]==x){cout<<i<<endl;k=1;break;}
if(!k)cout<<"Fail!"<<endl;
```

- 这种按顺序查找的方法称为**顺序查找**。顺序查找的特点是对任意序列都可实现查找，然而，当序列很大时，查找次数很多，显然有许多重复的操作，查找效率低。

【例1】查成绩 --1116



【问题描述】 中考成绩出来了，许多考生想知道自己成绩排名情况，于是考试委员会找到了你，让你帮助完成一个成绩查找程序，考生只要输入成绩，即可知道自己的排名。

【输入格式】

第1行两个整数N和K($N, K \leq 10000$);

第2行N个从大到小不相同的整数(小于等于 10^9), 表示考生的成绩;

第三行为K个待查找的考生成绩。


【输出格式】 输出共计K行，每行为一个待查找的考生的名次，即有多少个考生分数比他高。

【输入样例】


```
5 2
100 99 98 96 90
99 98
```

【输出样例】

```
2
4
```



■**思考：** 已知一个8000以内的正整数，让你去猜这个数，然后告诉你所猜的答案太大还是太小，直到你猜到为止。你会怎样去猜，使得猜的次数尽量少呢？



- 前提：数据是**有序**的，那么在这批数据中查找某一数据就可以用上述的二分查找实现。
- 二分查找算法描述如下，假设数据**从小到大排序好的**：
- 1) 需要设置三个指针L、R、mid表示查找区间的左端点、右端点和中间位置。
- 2) 当 $L \leq R$ 时，在a[]数组中二分查找X的过程为：
 - ①取 $mid = (L + R) / 2$;
 - ②若 X 等于a[mid]，返回查找结果并退出查找
 - ③若 X 大于a[mid]，那么 $L = mid + 1$,继续二分;
 - ④若 X 小于a[mid]，那么 $R = mid - 1$,继续二分;

核心代码

```
cin>>x;  
int L=1,R=n;  
while(L<=R)  
{   int mid=(L+R)/2;  
    if(x==a[mid]){cout<<mid<<endl;break;}  
    if(x>a[mid])L=mid+1;  
    else R=mid-1;  
}
```

【演示】查找值为22的记录过程



下标:

1	2	3	4	5	6	7	8	9	10
13	15	18	22	28	34	56	65	70	80
↑ L=1	↑ mid=2	↑ L=3	↑ R=4	↑ mid=5					↑ R=10

15 < 22 28 > 22

18 < 22

22 == 22, 查找成功

```
cin>>x;
int L=1,R=n;
while(L<=R)
{   int mid=(L+R)/2;
    if(x==a[mid])
        {cout<<mid<<endl;break;}
    if(x>a[mid])L=mid+1;
    else R=mid-1;
}
```


【例1】查成绩 --1116



【问题描述】 中考成绩出来了，许多考生想知道自己成绩排名情况，于是考试委员会找到了你，让你帮助完成一个成绩查找程序，考生只要输入成绩，即可知道自己的排名。

【输入格式】

第1行两个整数N和K($N, K \leq 10000$);

第2行N个从大到小不相同的整数(小于等于 10^9), 表示考生的成绩;

第三行为K个待查找的考生成绩。

【输出格式】 输出共计K行，每行为一个待查找的考生的名次，即有多少个考生分数比他高。

【输入样例】

```
5 2
100 99 98 96 90
99 98
```

【输出样例】

```
2
4
```

参考代码

```
int n,k,a[10005],i,j,x;
int main()
{  cin>>n>>k;
   for(i=1;i<=n;i++)cin>>a[i];
   for(i=1;i<=k;i++)
   {  cin>>x;
      int L=1,R=n,mid;
      while(L<=R)
      {  mid=(L+R)/2;
         if(x==a[mid]){cout<<mid<<endl;break;}
         if(x>a[mid])R=mid-1;
         else L=mid+1;
      }
   }
   return 0;
}
```