#include<iostream> // 二分查找的递归与非递归实现

using namespace std; // 分治法，可分，可合，子问题有独立性

int binarySearchLoop(int arr[], int len, int findData)

{

if(arr==NULL || len <=0)

return -1;

int start = 0;

int end = len-1;

while(start<=end)

{

int mid = start+(end-start)/2;

if(arr[mid] == findData)

return mid;

else if(findData < arr[mid])

end = mid-1;

else

start = mid+1;

}

return -1;

}

//递归有自调用的问题，需要将start和end写在参数列表中来标记和动态变更搜索范围的开始和结束

int binarySearchRecursion(int arr[], int findData, int start, int end)

{

if(arr==NULL || start>end)

return -1;

int mid = start+(end-start)/2;

if(arr[mid] == findData)

return mid;

else if(findData < arr[mid])

binarySearchRecursion(arr, findData, start, mid-1);

else

binarySearchRecursion(arr, findData, mid+1, end);

}

void main()

{

int arr[] = {1,2,3,4,5,6,7,8};

int len = sizeof(arr)/sizeof(arr[0]);

int index = binarySearchLoop(arr,len,6); // 5

int index2 = binarySearchRecursion(arr,6,0,len-1); // 5

cout<<index<<endl;

cout<<index2<<endl;

system("pause");

}