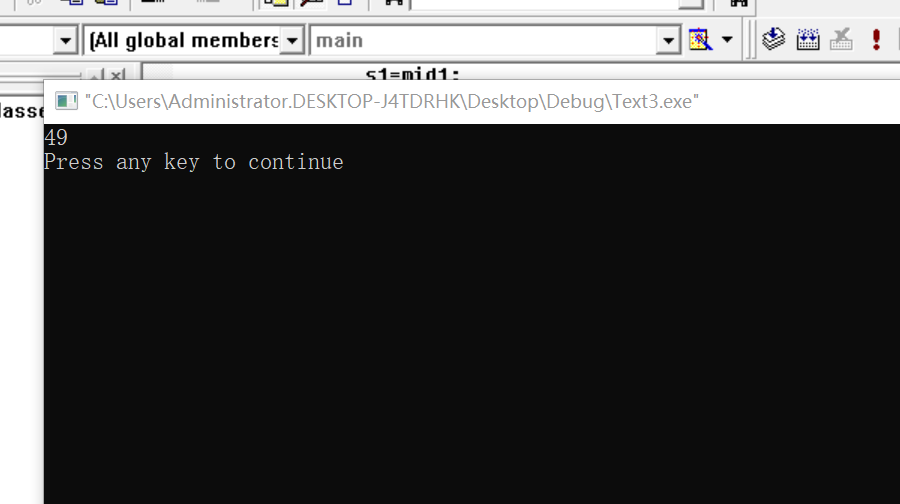
#include<stdio.h>  
#include <iostream>  
#include<iomanip>  
using namespace std;  
  
int SearchMId(int A[],int B[],int n){  
int s1=0,e1=n-1,s2=0,e2=n-1;  
int mid1,mid2;  
while(s1<e1&&s2<e2){  
mid1=(s1+e1)/2;  
mid2=(s2+e2)/2;  
if(A[mid1]==B[mid2])  
return A[mid1];  
if(A[mid1]<B[mid2]){  
if((s1+e1)%2==0)  
s1=mid1;  
else  
s1=mid1+1;  
e2=mid2;  
}  
else {  
if((s2+e2)%2==0)  
s2=mid2;  
else  
s2=mid2+1;  
e1=mid1;  
}  
}  
if(A[s1]<B[s2])  
return A[s1];  
else  
return B[s2];  
}  
  
void main(){  
int a[]={10,20,23,25,63,89,90,95,100};  
int b[]={6,13,15,45,49,52,59,77,55};  
int MId=SearchMId(a,b,9);  
cout<<MId<<endl;  
}



#include<stdio.h>  
#include <iostream>  
#include<iomanip>  
using namespace std;  
  
int BinSearch1(int r[],int n,int k){  
int low=0;  
int high=n-1;  
int mid;  
while(low<=high)  
{  
mid=(low+high)/2;  
        if(k<r[mid])  
high=mid-1;  
else if(k>r[mid])  
low=mid+1;  
else  
return mid;  
}  
return -1;  
}  
  
void main(){  
  
int a[]={10,23,54,65,68,69,70,75,79,80,99};  
int m=BinSearch1(a,11,70);  
cout<<"第"<<m+1<<"个位置"<<endl;  
}



#include<stdio.h>  
#include <iostream>  
#include<iomanip>  
using namespace std;  
struct BiNode{  
int data;  
BiNode \*lchild,\*rchild;  
};  
BiNode \* SearchBST(BiNode \* root,int k)  
{  
if(root==NULL)  
return NULL;  
else if(root->data==k)  
return root;  
else if(k<root->data)  
return SearchBST(root->lchild,k);  
else  
return SearchBST(root->rchild,k);  
}  
  
BiNode \*InsertBST(BiNode \* root,int data)   //插入二叉树  
{  
if(root==NULL)  
{  
root=new BiNode;  
root->data=data;  
root->lchild=root->rchild=NULL;  
return root;  
}  
if(data<=root->data)  
root->lchild=InsertBST(root->lchild,data);  
else  
        root->rchild=InsertBST(root->rchild,data);  
return root;  
}  
  
BiNode \*createBST(int a[],int n)   //新建二叉树  
{  
BiNode \*T=NULL;  
for(int i=0;i<n;i++){  
T=InsertBST(T,a[i]);   //插入数据  
}  
return T;  
}  
  
void main(){  
int a[]={12,5,11,45,84,1,51,89,16,48};  
      
BiNode \* D=NULL;  
D=createBST(a,10);  
if(SearchBST(D,5)!=NULL)   //查找的数据在SearchBST（）方法的第二个参数    
cout<<"找到需要查找的数字"<<endl;  
else  
cout<<"找不到需要查找的数字"<<"5"<<endl;  
}

