#include<iostream>

using namespace std;

#include<stdlib.h>

#define MAXSIZE 100

#define OK 1

#define ERROR 0

#define OVERFLOW -2

#define MAXSIZE 100

typedef int ElemType;

typedef int Status;

typedef struct

{

ElemType \*elem;

int length;

}SqList;

//初始化

Status InitList(SqList &L);

//输入

Status Input(SqList &L,int i,int e);

//输出

void Output(SqList &L,int i);

//取值

Status GetElem(SqList L,int i,ElemType &e);

/\*int LocateElem(SqList L,ElemType e);//查找

\*/

//排序

Status sort(SqList &L);

//有序插入

Status Fun(SqList &L,int e);

//插入

Status ListInsert(SqList &L,int i,ElemType e);

//删除

Status ListDelete(SqList &L,int i);

//---------------------------------------------------------------------------------------

int main()

{

int i;

SqList L;

ElemType e;

//输入-------------------------

InitList(L);

cout<<"输入：";

for(i=0;i<6;i++)

{

cin>>e;

Input(L,i,e);

}

//输出-------------------------

cout<<"输出：";

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

//-----------------------------

i=1,e=3;

if(ListInsert(L,i,e)==OK)

{

cout<<"插入成功！"<<endl;

cout<<"结果为：";

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

}

else cout<<"插入失败"<<endl;

i=8,e=21;

if(ListInsert(L,i,e)==OK)

{

cout<<"插入成功！"<<endl;

cout<<"结果为：";

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

}

else cout<<"插入失败"<<endl;

i=4,e=15;

if(ListInsert(L,i,e)==OK)

{

cout<<"插入成功！"<<endl;

cout<<"结果为：";

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

}

else cout<<"插入失败"<<endl;

i=12,e=99;

if(ListInsert(L,i,e)==OK)

{

cout<<"插入成功！"<<endl;

cout<<"结果为：";

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

}

else cout<<"插入失败"<<endl;

//------------------------------

i=1;

if(ListDelete(L,i)==OK)

{

cout<<"删除成功！"<<endl;

cout<<"结果为：";

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

}

else cout<<"删除失败"<<endl;

i=9;

if(ListDelete(L,i)==OK)

{

cout<<"删除成功！"<<endl;

cout<<"结果为：";

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

}

else cout<<"删除失败"<<endl;

i=12;

if(ListDelete(L,i)==OK)

{

cout<<"删除成功！"<<endl;

cout<<"结果为：";

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

}

else cout<<"删除失败"<<endl;

//------------------------------

sort(L);

cout<<"排序后结果为："<<endl;//排序

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

//-------------------------------

cout<<"有序插入一个数："<<endl;

Fun(L,20); //有序插入 20

cout<<"插入后结果为："<<endl;

for(i=0;i<L.length;i++) Output(L,i);

cout<<endl;

Fun(L,50); //有序插入 50

cout<<"插入后结果为："<<endl;

for(i=0;i<L.length;i++) Output(L,i);

//--------------------------------

return 0;

}

//初始化

Status InitList(SqList &L)

{

L.elem=new ElemType[MAXSIZE];

if(!L.elem) return ERROR;

L.length=0;

return OK;

}

//输入

Status Input(SqList &L,int i,int e)

{

if(i>L.length||i<0) return ERROR;

L.elem[i]=e;

++L.length;

return OK;

}

//输出

void Output(SqList &L,int i)

{

cout<<L.elem[i]<<" ";

}

//取值

Status GetElem(SqList L,int i,ElemType &e)

{

if(i<1||i>L.length) return ERROR;

e=L.elem[i-1];

return OK;

}

//插入

Status ListInsert(SqList &L,int i,ElemType e)

{

int j;

if((i<1)||(i>L.length+1)) return ERROR;

if(L.length==MAXSIZE) return ERROR;

for(j=L.length-1;j>=i-1;j--)

L.elem[j+1]=L.elem[j];

L.elem[i-1]=e;

++L.length;

return OK;

}

//删除

Status ListDelete(SqList &L,int i)

{

if((i<1||i>L.length)) return ERROR;

for(int j=1;j<=L.length;j++)

L.elem[j-1]=L.elem[j];

--L.length;

return OK;

}

//排序

Status sort(SqList &L)

{

int i,j,t;

for(i=0;i<L.length-1;i++)

{

for(j=0;j<L.length-1-i;j++)

{

if(L.elem[j]>L.elem[j+1])

{

t=L.elem[j];

L.elem[j]=L.elem[j+1];

L.elem[j+1]=t;

}

}

}

}

//有序插入

Status Fun(SqList &L,int e)

{

int i,flag;

for(i=0;i<L.length;i++)

{

if(e>=L.elem[i]) flag=i+1;

}

ListInsert(L,flag+1,e);

return OK;

}