```
/*第3次作业
实现图书信息管理，要求：
设计图书顺序表结构
typedef struct{
 char bname[100];
 char bno[100];
 …//根据功能需要补充
}BOOK;
typedef{BOOK *elem;int len; int listsize;}
功能函数：
1、初始化创建一个空表；
2、任意位置插入；
3、任意位置删除；
4、按书名进行查找；
5、按指定位置查找；
7、求顺序表的长度；
8、修改某条记录的某个成员；
9、输出顺序表。

主函数：定义顺序表变量，通过调用各功能函数顺序完成以下操作：
初始化空表，连续插入3条数据到表头，输出，
插入一条数据使其成为表中第3条数据，输出，
删除第3条数据，输出，
按书号进行查找，求表长，
按书名查找后进行对应记录书号的修改，输出。
*/
#define OK 1
#define ERROR 0

#define MAXSIZE 100

typedef int status;
```

```
typedef struct
{
    char bno[10];
    char bname[10];
    float price;
}BOOK;

typedef struct{
    BOOK *elem;
    int len;
    int listsize;
}SqList;

status InitList(SqList*); //初始化创建空表
status InsertList(SqList *,int, BOOK); //插入元素到表中的指定位置
status DeleteList(SqList *,int, BOOK*);//任意位置删除
status CreatList(SqList*,int ); //连续插入一些元素到表头
status PrintList(const SqList*); //打印输出表中的元素
int GetLength(SqList L);
int GetByName(SqList L,char name[]);//按书名查找，返回位序
status GetByPos(SqList ,int,BOOK* );
status UpdateElem(SqList, char no[],float);//按no进行查找修改价格

//主函数进行测试,输入数据是:
3
1 1 1
2 2 2
3 3 3
4 4 4 1
int _tmain(int argc, _TCHAR* argv[])
{
```

```
SqList L;
if(InitList(&L) == ERROR)
    printf("初始化创建表失败！\n");
else
    printf("初始化创建表成功！\n");
PrintList(&L);
int n;
printf("输入n个元素到表中\n");
scanf("%d",&n);
if(CreatList(&L,n))
    printf("\n插入了%d个元素到表头",n);
PrintList(&L);

//输入插入位置及元素
BOOK e;
printf("\n输入插入元素和位置\n");
scanf("%s%s%f%d",e.bno,e.bname,&e.price,&n);
if(InsertList(&L,n,e)==ERROR)
   printf("插入失败！\n");
else
    PrintList(&L);

DeleteList(&L,1,&e);
PrintList(&L);
DeleteList(&L,L.len,&e);
PrintList(&L);
int j = GetByName(L,"2");
GetByPos(L,2,&e);
UpdateElem(L,"2",222);
PrintList(&L);

return 0;
```

```
}
status UpdateElem(SqList L,char no[],float price)
{
    if(L.len == 0)
    {
        printf("空表！\n");
        return ERROR;
    }
    int i;
    for(i=0;i<L.len;i++)
        if(strcmp(L.elem[i].bno,no)==0)
            break;
    if(i<L.len)
        {
            L.elem[i].price = price;
            printf("修改成功！\n");
            return OK;
    }
    else
    {
        printf("没有该条记录！\n");
        return ERROR;
    }

}

status GetByPos(SqList L,int pos,BOOK *e)
{
    if(pos<1 || pos>L.len)
    {
        printf("给定位置不合理！\n");
```

```
        return ERROR;
    }
    if(L.len == 0)
    {   printf("表为空！\n");
        return ERROR;
    }
    printf("查找成功！\n");
    *e = L.elem[pos-1];
    return OK;
}

int GetByName(SqList L,char name[]) //返回位序
{
    int i;

    for(i=0;i<L.len;i++)
        if(strcmp(L.elem[i].bname,name)==0)
            break;
    if(i<L.len)
        return i;
    else return 0;
}
status DeleteList(SqList *L,int pos,BOOK *e)
{
    if(pos<1 || pos>L->len)
    {
        printf("删除位置不合理！\n");
        return ERROR;
    }
    if(L->len == 0)
    {
        printf("空表！\n");
```

```
        return ERROR;
    }
   *e = L->elem[i-1];
    int i;
    for(i=pos;i<L->len;i++)
        L->elem[i-1] = L->elem[i];
    L->len--;
    printf("删除成功！\n");
    return OK;
}
int GetLength(SqList L)
{
    return L.len;
}

status PrintList(const SqList *L)
{
    if(L->len == 0)
    {
       printf("\n表为空！\n");
       return ERROR;
    }
    printf("书号，书名，价格\n");
    for(int i=0; i<L->len; i++)
        printf("%s\t%s\t%f\n",L->elem[i].bno,L->elem[i].bname,L->elem[i].price);

}
status CreatList(SqList *L,int n)
{
    if(n==0)
    {
        printf("输入n不应为0！\n");
```

```
        return ERROR;
    }
    printf("输入%d个书号，书名，价格：",n);
    for(int i=0; i<n; i++)
        scanf("%s%s%f",L->elem[i].bno,L->elem[i].bname,&L->elem[i].price);
    L->len = n;
    return OK;
}
status InsertList(SqList *L, int pos, BOOK e)
{
    if(L->len == L->listsize)
    {
        printf("表满！\n");
        return ERROR;}
    if(pos<1 || pos>L->len+1)
    {
        printf("输入的插入位置不合理！\n");
        return ERROR;
    }

    int i;
    for(i=L->len;i>=pos;i--) //移动元素
        L->elem[i] = L->elem[i-1];
    L->elem[i] = e;
    L->len++;

    printf("插入成功！\n");
    return OK;
}

status InitList(SqList *L)
{
```

```
    L->elem = NULL;
    L->elem = (BOOK*)malloc(sizeof(BOOK*)*MAXSIZE);
    if(!L->elem)
        return ERROR;
    L->len = 0;
    L->listsize =MAXSIZE;
    return OK;
}
```