**数据结构** ——线性表的原理及c++程序实现

什么是线性表？

线性表：由同类型数据元素构成有序序列的线性结构

>表中元素个数称为线性表的长度

>线性表没有元素时，称为空表

>表起始位置称表头，表结束位置称表尾

线性表的抽象数据类型描述

类型名称：线性表（**List**）

数据对象集：线性表是 **n (≥0)**个元素构成的有序序列**( a1, a2, …,an )**

操作集：线性表**L** ∈ **List**，整数**i**表示位置，元素**X** ∈ **ElementType**，

线性表基本操作主要有：

**1**、**List MakeEmpty()**：初始化一个空线性表**L**；

**2**、**ElementType FindKth( int K, List L )**：根据位序**K**，返回相应元素 ；

**3**、**int Find( ElementType X, List L )**：在线性表**L**中查找**X**的第一次出现位置；

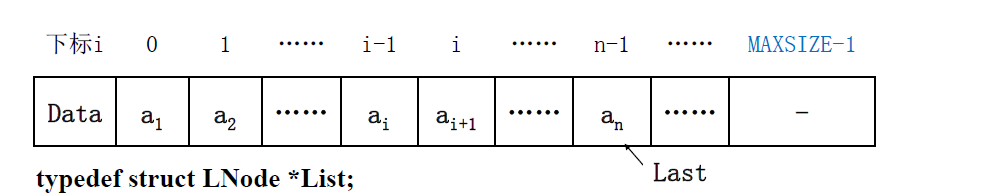
**4**、**void Insert( ElementType X, int i, List L)**：在位序**i**前插入一个新元素**X**；

**5**、**void Delete( int i, List L )**：删除指定位序**i**的元素；

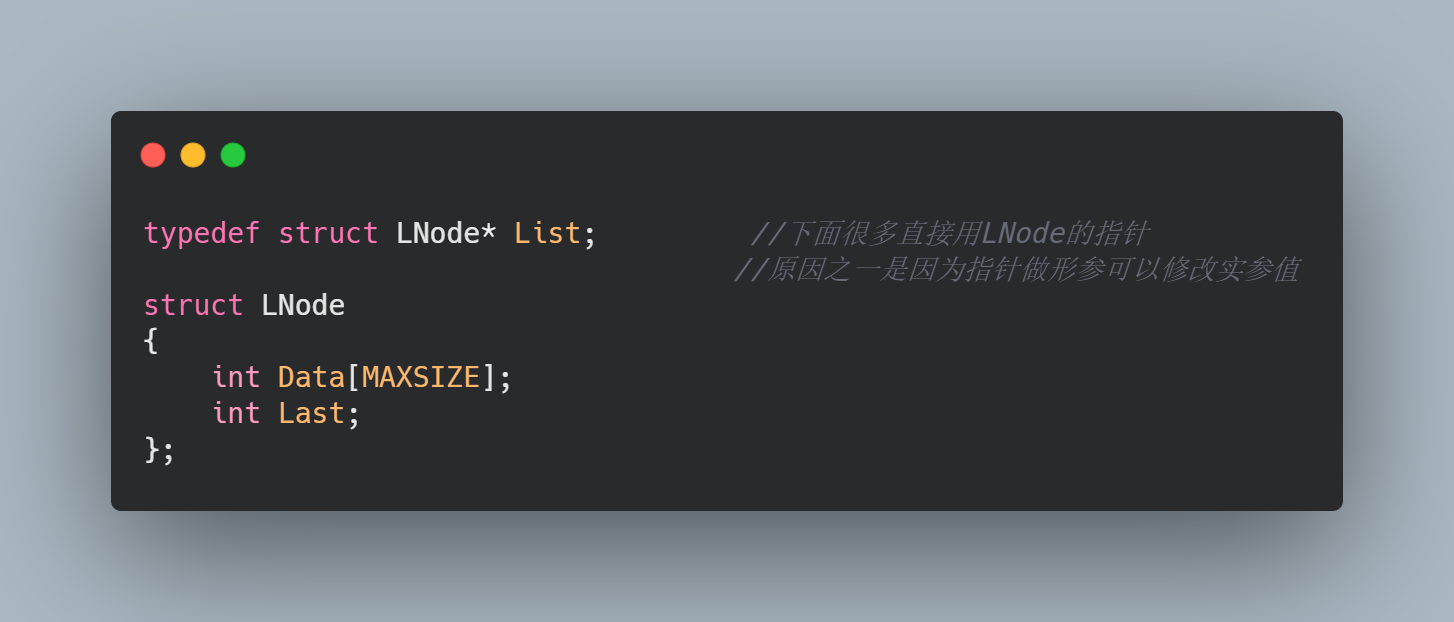
**6**、**int Length( List L )**：返回线性表**L**的长度**n**。

线性表的顺序存储实现：

利用数组的连续存储空间顺序存放线性表的各元素

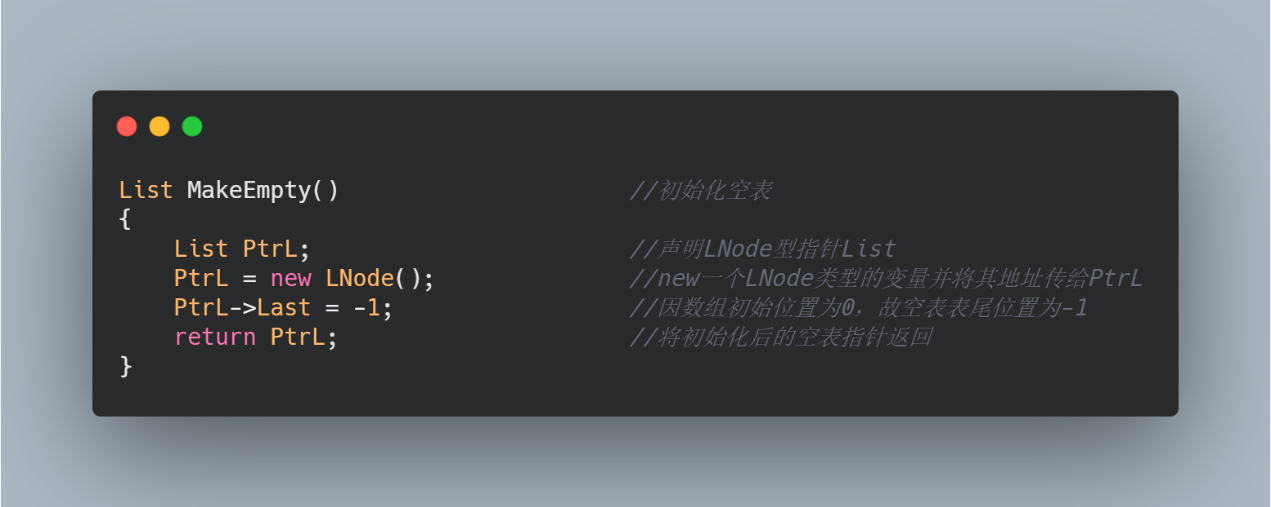


定义线性表的数据对象集

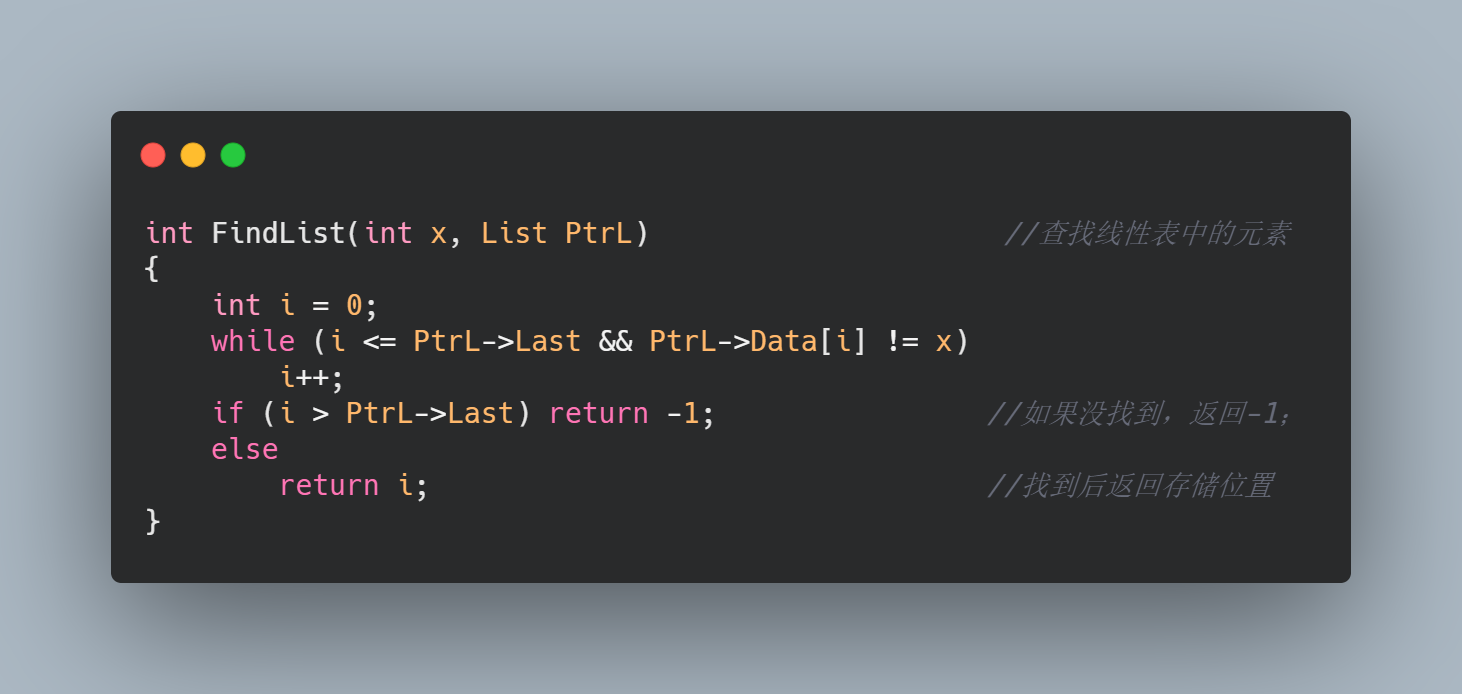


定义线性表的操作集

**初始化空表：**

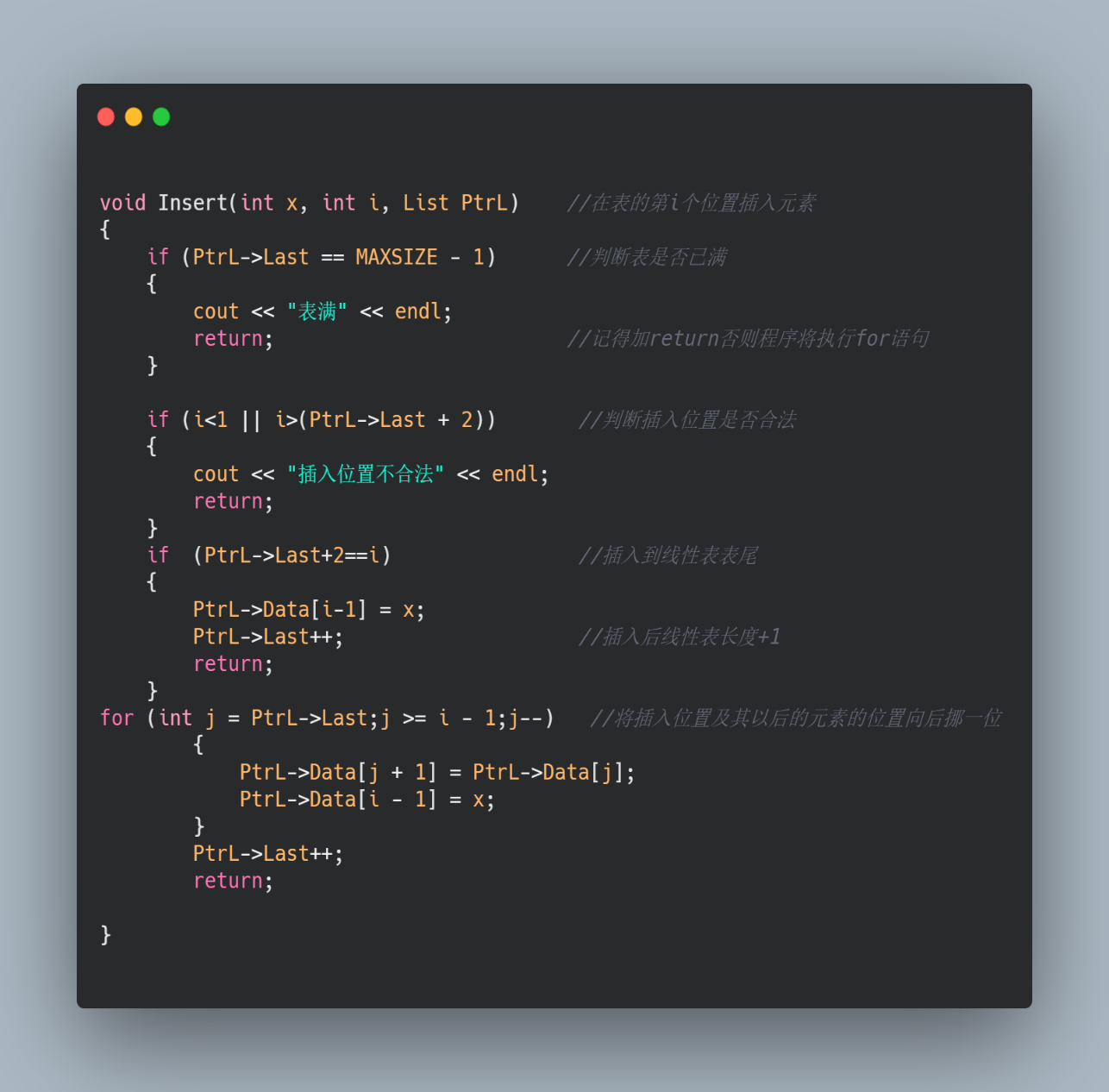


**查找线性表中的元素**

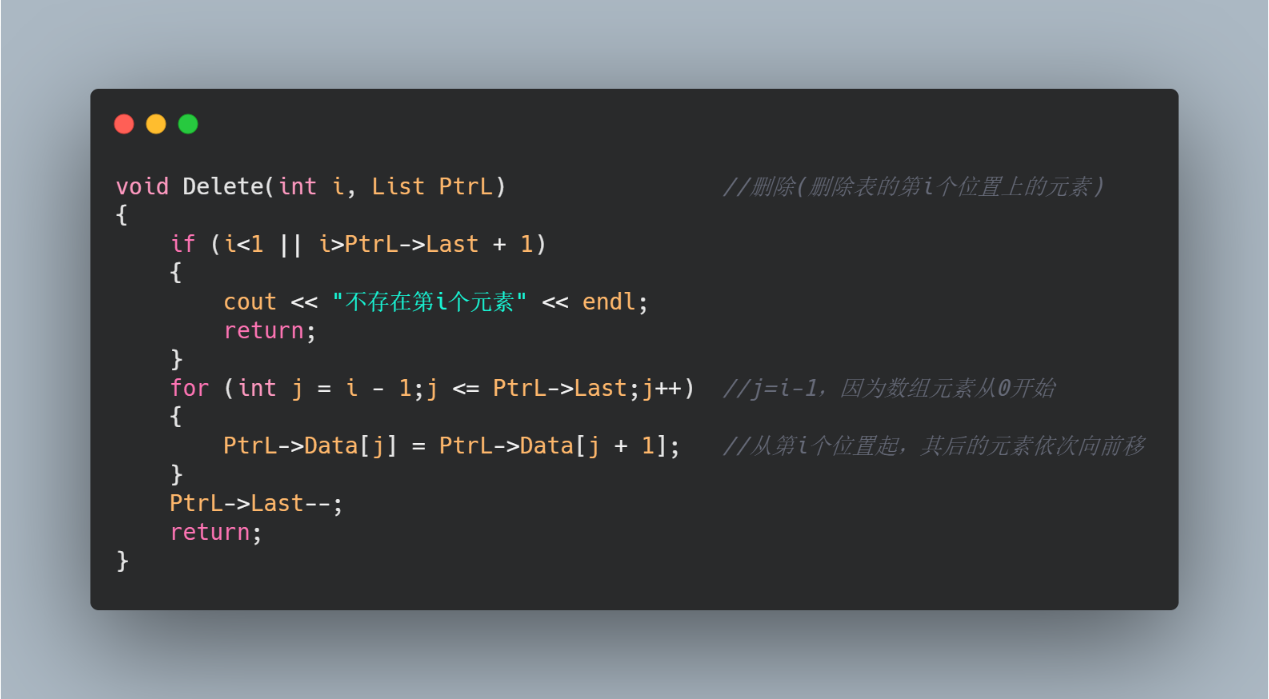


**向线性表中插入元素**

PS:相较于老师在视频中写的插入操作，我自己增加了插入到线性表尾的操作，可以用来初始化线性表



**删除线性表中的元素**

****

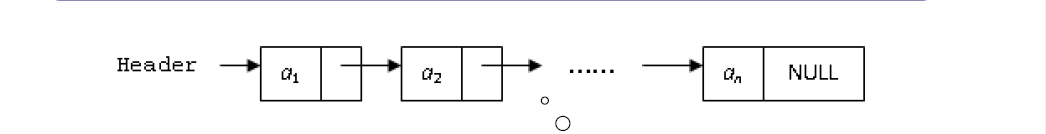
**线性表的完整代码如下**

****

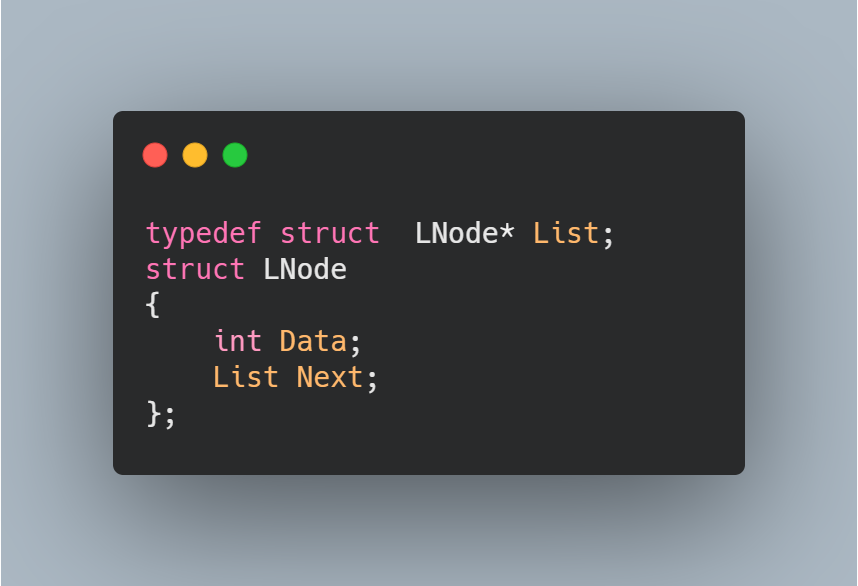
线性表的链式存储实现：

不要求逻辑上相邻的两个元素物理上也相邻；通过“链”建立起数据元素之间的逻辑关系

>插入，删除不需要移动数据元素，只需要修改“链”



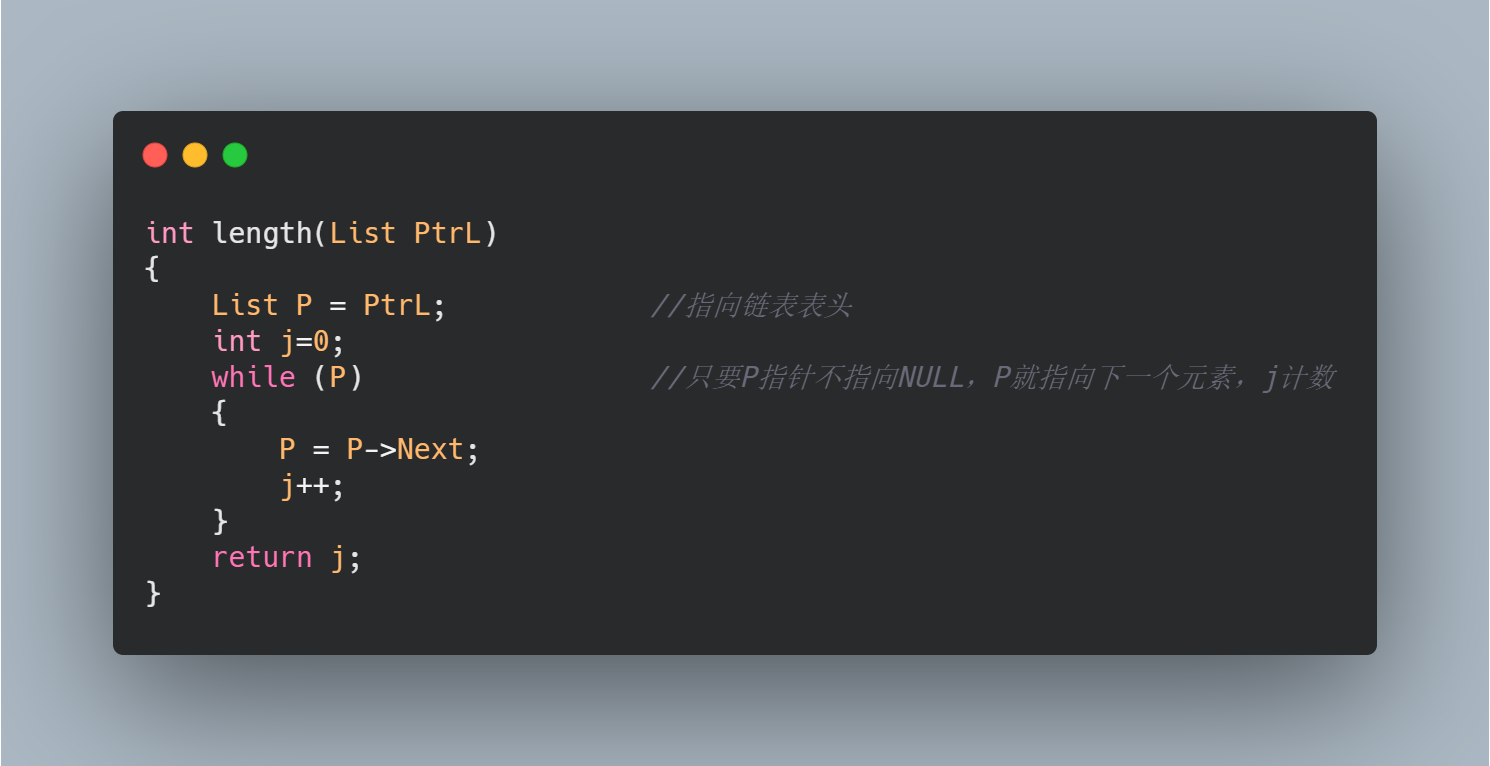
**定义链表的数据对象集**



链表的每一个结点都代表一个结构，一个结构至少有两个分量，一个分量代表这个结点对应的数据，另一个代表它的下一个结点的位置

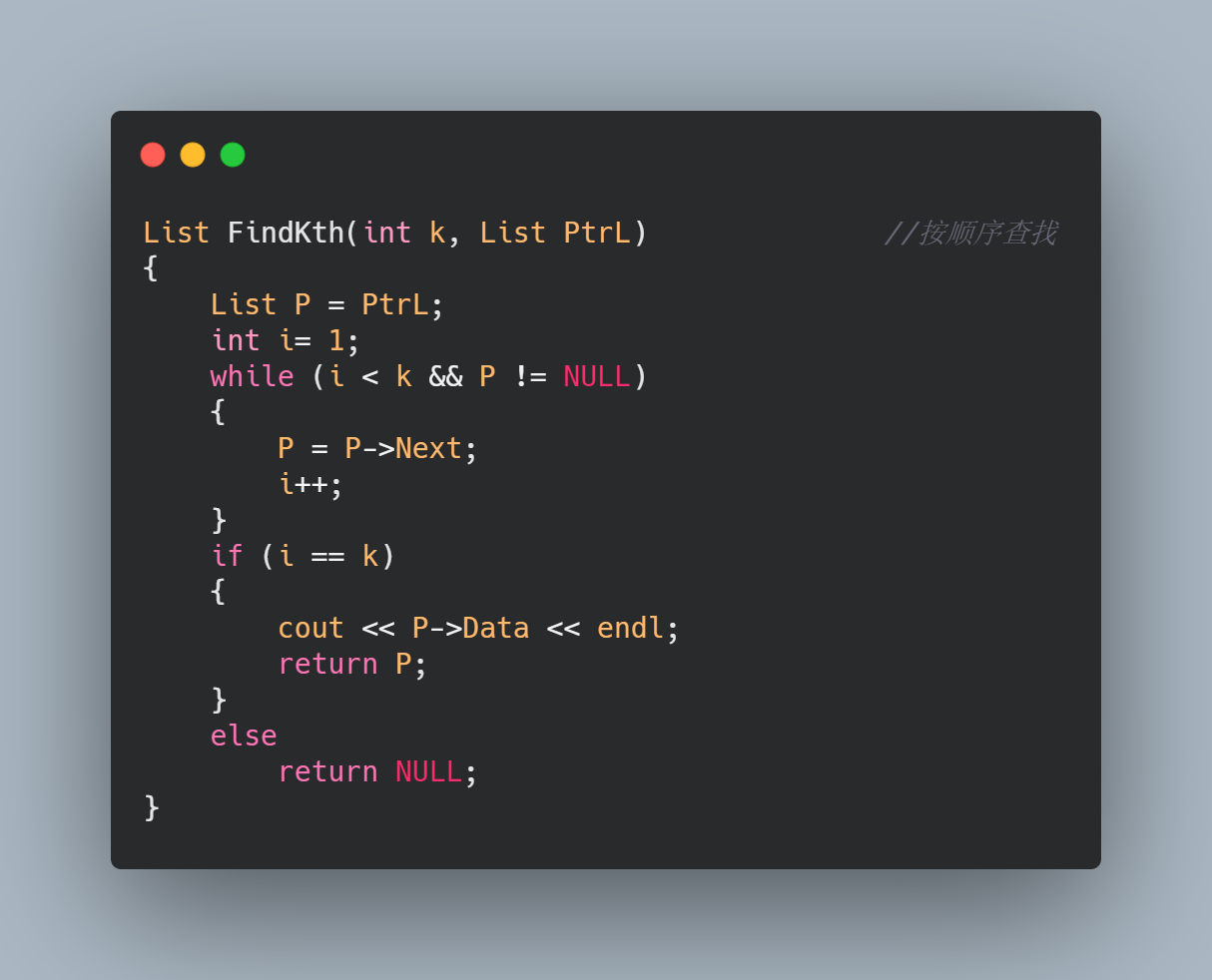
**定义链表的操作集**

**查表长**



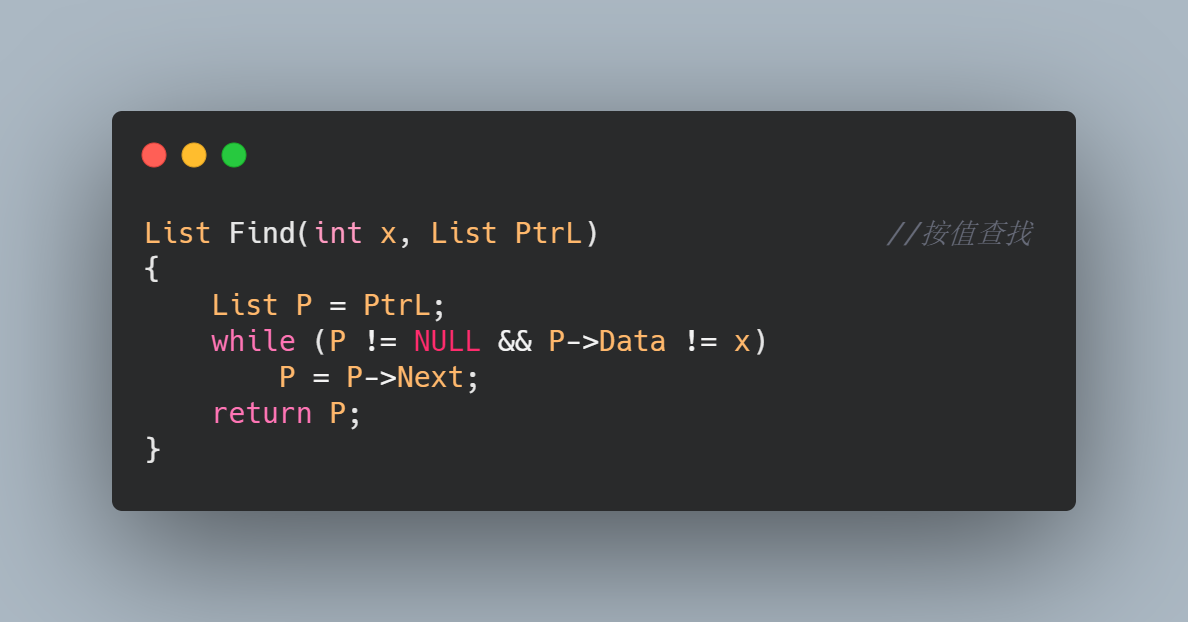
首先将指针指向链表表头，计数器定为0，将指针依次后移直到指向空指针，计数器依次+1

**按序号查找**

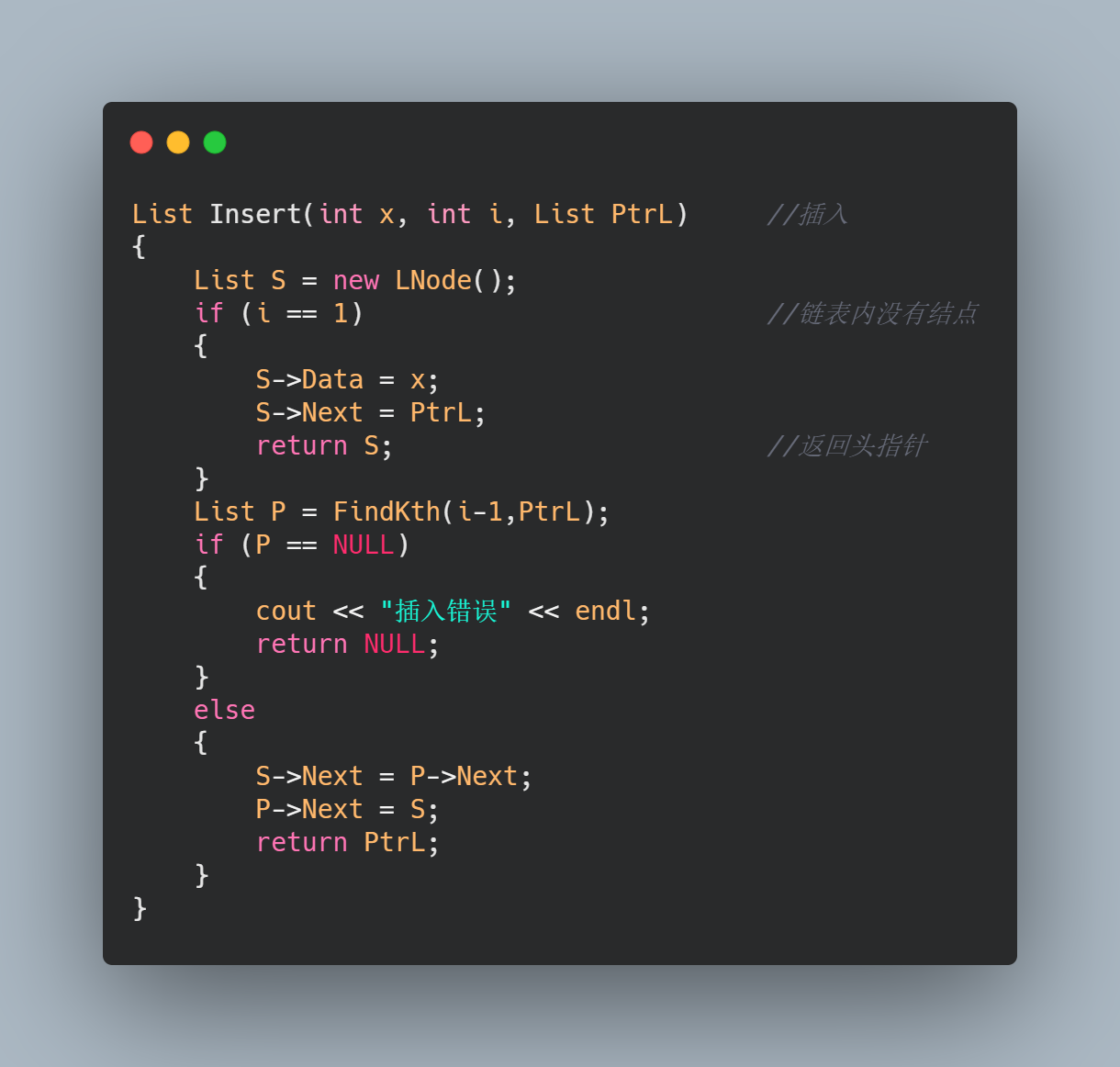


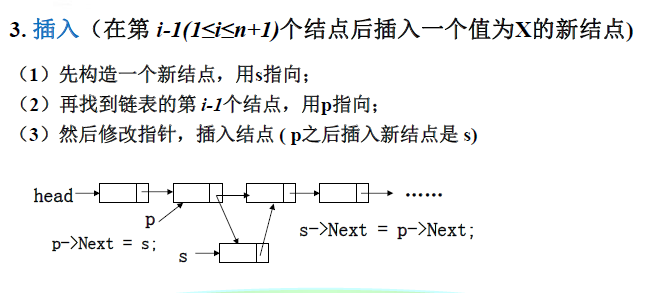
链表寻找元素不同于顺序表，顺序表逻辑相邻的元素顺序也相邻，故数组下标+1即可指向下一元素，知道逻辑位置即可知道物理位置，链表依靠指针NEXT连接在一起，如果想要知道某逻辑位置的值必须依靠指针从头依次遍历找到其物理位置，返回物理位置。

**按值查找**

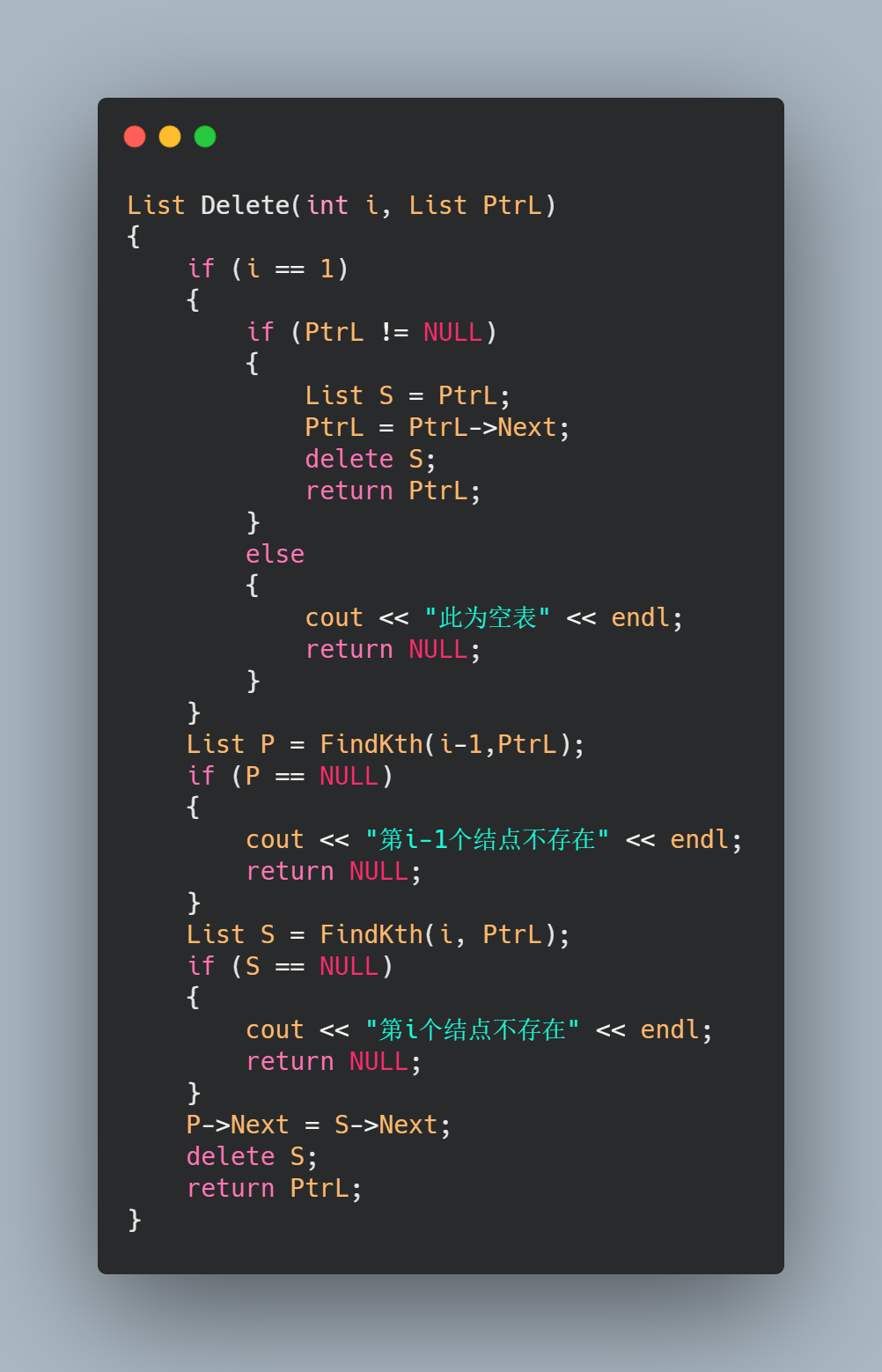


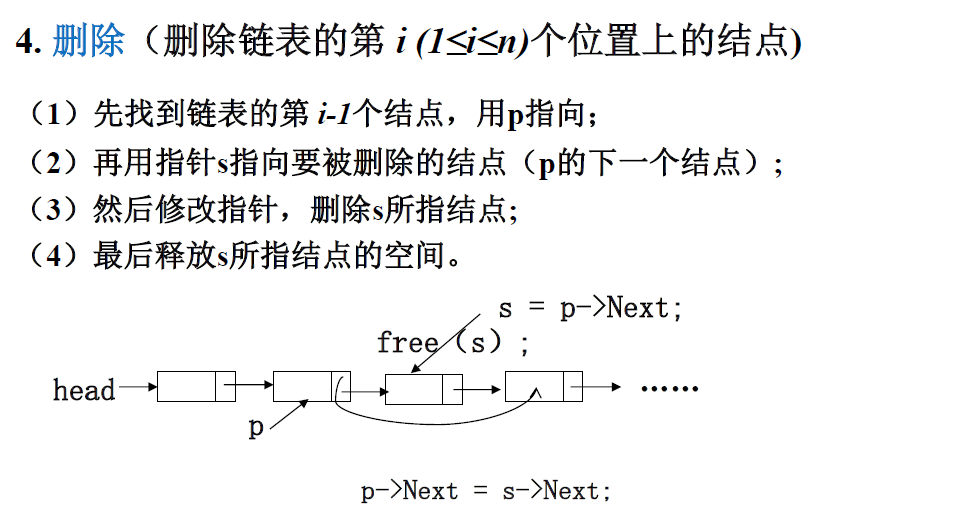
**链表的插入**

****



**链表删除**

****

****

由于链表中的每一个结点都是通过new函数申请得到的，故删除以后为防止内存泄露需要将其delete掉。

删除头结点有两种情况，一种头结点本身为空，此时返回NULL即可，一种是头结点本身不空，则将一个指针指向要被删除的结点，将头结点后移一位，然后删除指针所指向的结点，此时表空

完整代码

****