

题目一 在 $O(1)$ 时间内删除链表节点

题目一：在 $O(1)$ 时间内删除链表节点。

给定单向链表的头指针和一个节点指针，定义一个函数在 $O(1)$ 时间内删除该节点。链表节点与函数的定义如下：

解答

方法：假设要删除节点 i ，他的下一个节点是 j ，先把 j 的值复制给 i ，然后 i 的 `next` 指向 j 的下一个节点，然后删除 j 。

分三种情况：

1. 要删除的不是尾结点
2. 链表只有一个节点，删除它
3. 链表中有多个节点，删除的是尾结点

```
void DeleteNode(ListNode** head, ListNode* pdelete)
{
    if(!head || !pdelete)
        return;
    if(pdelete->next)//要删除的不是尾结点
    {
        ListNode* pnext=pdelete->next;
        pdelete->val=pnext->val;
        pdelete->next=pnext->next;
        delete pnext;
        pnext=nullptr;
    }
    else if(*head==pdelete)
    {
        delete pdelete;
        pdelete=nullptr;
        *head=nullptr;
    }
    else//多个节点，删除尾结点
    {
        ListNode* node=*head;
        while(node->next != pdelete)
        {
            pnode=pnode->next;
        }
        pnode->next=nullptr;
        delete pdelete;
        pdelete=nullptr;
    }
}
```

```
}  
}
```

题目二 删除链表中重复的节点

```
ListNode* deleteDuplication(ListNode* phead)  
{  
    if(!phead)  
        return nullptr;  
    if(!phead->next)  
        return phead;  
    ListNode* dummy=new ListNode(0),*pre=dummy;  
    dummy->next=phead;  
    ListNode* pnext=nullptr;  
    ListNode* node=phead;  
    while(node && node->next)  
    {  
        pnext=node->next;  
        if(node->val==pnext->val)  
        {  
            while(node->val == pnext->val && pnext)  
            {  
                pnext=pnext->next;  
            }  
            pre->next=pnext;  
            node=pnext;  
        }  
        else  
        {  
            pre=node;  
            node=node->next;  
        }  
    }  
    return dummy->next;  
    delete dummy;  
    dummy=nullptr;  
}
```