

C\_C 中结构体(struct)知识点强化 (三) PDF转换可能丢失图片或格式, 建议阅读原文

[https://www.100test.com/kao\\_ti2020/136/2021\\_2022\\_C\\_C\\_\\_\\_E4\\_B8\\_AD\\_E7\\_BB\\_c97\\_136115.htm](https://www.100test.com/kao_ti2020/136/2021_2022_C_C___E4_B8_AD_E7_BB_c97_136115.htm) 下面我们来说一下链表节点的删除!

我们以上面的程序为基础, 但为了我们方便学习删除我们调整结构体为 `struct test { int number. float socre. test *next. }`. `number` 为唯一的编号每一个节点的。删除的我就不多说了, 里面重要部分有注解。特别注意 `0deletel` 函数的参数意义, 指针的引用在这里很重要, 如果只是指针, 或者只是应用都是不行的, 为什么仔细思考, 很多知名的教材在这一问题上都很模糊, 而且很多书还有错误, 程序不错, 但思路是错的, 我这里特别不说, 请大家仔细阅读程序, 如果还是有问题, 可以回此帖, 我会回答的。来源: [www.examda.com](http://www.examda.com) 完整代码如下:

```
#include using namespace std. struct test { int number. float socre. test *next. }. test *head.//创建一个全局的引导进入链表的指针 test *create() { test *ls.//节点指针 test *le.//链尾指针 ls = new test.//把ls指向动态开辟的堆内存地址
```

```
cin>>ls->number>>ls->socre. head=NULL.//进入的时候先不设置head指针指向任何地址,因为不知道是否一上来就输入null跳出程序 le=ls.//把链尾指针设置成刚刚动态开辟的堆内存地址,用于等下设置le->next,也就是下一个节点的位置
```

```
while(ls->number!=0)//创建循环条件为ls->number的值不是null,用于循环添加节点 { if(head==NULL)//判断是否是第一次进入循环 { head=ls.//如果是第一次进入循环,那么把引导进入链表的指针指向第一次动态开辟的堆内存地址 } else { le->next=ls.//如果不是第一次进入那么就上一次链尾指针
```

```
的le->next指向上一次循环结束前动态创建的堆内存地址 }  
le=ls;//设置链尾指针为当前循环中的节点指针,用于下一次进入循环的时候把上一次的节点的next指向上一次循环结束前动态创建的堆内存地址 ls=new test;//为下一个节点在堆内存中动态开辟空间 cin>>ls->number>>ls->socre. } le->next=NULL;//把链尾指针的next设置为空,因为不管如何循环总是要结束的,设置为空才能够在循环显链表的时候不至于死循环 0delete ls;//当结束的时候最后一个动态开辟的内存是无效的,所以必须清除掉 return head;//返回链首指针 } void showl(test *head) { cout while(head)//以内存指向为null为条件循环显示先前输入的内容 来源 : www.examda.com { cout head=head->next. } } 100Test  
下载频道开通 , 各类考试题目直接下载。详细请访问  
www.100test.com
```