

C语言中的typedef问题 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/220/2021_2022_C_E8_AF_AD_E8_A8_80_E4_B8_AD_c67_220285.htm

1. 基本解释 typedef 为C语言的关键字，作用是作为一种数据类型定义一个新名字。这里的数据类型包括内部数据类型（int,char等）和自定义的数据类型（struct等）。在编程中使用typedef目的一般有两个，一个是给变量一个易记且意义明确的新名字，另一个是简化一些比较复杂的类型声明。至于typedef有什么微妙之处，请你接着看下面对几个问题的具体阐述。

2. typedef & struct 的问题 当用下面的代码定义一个结构时，编译器报了一个错误，为什么呢？莫非C语言不允许在结构中包含指向它自己的指针吗？请你先猜想一下，然后看下文说明：
`typedef struct tagNode { char *pItem. pNode pNext. } *pNode.`

答案与分析：
1、typedef的最简单使用 `typedef long byte_4.` 给已知数据类型long起个新名字，叫byte_4。
2、typedef与结构结合使用 `typedef struct tagMyStruct { int iNum. long lLength. } MyStruct.` 这语句实际上完成两个操作：1) 定义一个新的结构类型 `struct tagMyStruct { int iNum. long lLength. }.` 分析：tagMyStruct称为“tag”，即“标签”，实际上是一个临时名字，struct关键字和tagMyStruct一起，构成了这个结构类型，不论是否有typedef，这个结构都存在。我们可以用`struct tagMyStruct varName`来定义变量，但要注意，使用`tagMyStruct varName`来定义变量是不对的，因为struct和tagMyStruct合在一起才能表示一个结构类型。

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