

C语言笔记第五章数组和指针的关系 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_AF_AD_E8_A8_80_E7_AC_94_c97_135129.htm 第五章 数组和指针的关系 如果对声明数组的语句不太明白的话，请参阅我前段时间贴出的文章理解c和c的复杂类型声明>>。数组的数组名其实可以看作一个指针。看下例：例八：int array[10]={0,1,2,3,4,5,6,7,8,9},value. ... value=array[0].//也可写成：value=*array. value=array[3].//也可写成：value=*(array 3). value=array[4].//也可写成：value=*(array 4). 上例中，一般而言数组名array代表数组本身，类型是int [10]，但如果把array看做指针的话，它指向数组的第0个单元，类型是int *，所指向的类型是数组单元的类型即int。因此*array等于0就一点也不奇怪了。同理，array 3是一个指向数组第3个单元的指针，所以*(array 3)等于3。其它依此类推。例九：例九：char *str[3]={ "Hello,this is a sample!", "Hi,good morning.", "Hello world" }. char s[80]; strcpy(s,str[0]).//也可写成strcpy(s,*str). strcpy(s,str[1]).//也可写成strcpy(s,*(str 1)). strcpy(s,str[2]).//也可写成strcpy(s,*(str 2)). 上例中，str是一个三单元的数组，该数组的每个单元都是一个指针，这些指针各指向一个字符串。把指针数组名str当作一个指针的话，它指向数组的第0号单元，它的类型是char**，它指向的类型是char *。*str也是一个指针，它的类型是char*，它所指向的类型是char，它指向的地址是字符串"Hello,this is a sample!"的第一个字符的地址，即H的地址。str 1也是一个指针，它指向数组的第1号单元，它的类型是char**，它指向的类型是char *。*(str 1)也是一个指针

，它的类型是char*，它所指向的类型是char，它指向"Hi,good morning."的第一个字符H，等等。下面总结一下数组的数组名的问题。声明了一个数组TYPE array[n]，则数组名称array就有了两重含义：第一，它代表整个数组，它的类型是TYPE [n]；第二，它是一个指针，该指针的类型是TYPE*，该指针指向的类型是TYPE，也就是数组单元的类型，该指针指向的内存区就是数组第0号单元，该指针自己占有单独的内存区，注意它和数组第0号单元占据的内存区是不同的。该指针的值是不能修改的，即类似array的表达式是错误的。在不同的表达式中数组名array可以扮演不同的角色。在表达式sizeof(array)中，数组名array代表数组本身，故这时sizeof函数测出的是整个数组的大小。在表达式*array中，array扮演的是指针，因此这个表达式的结果就是数组第0号单元的值。sizeof(*array)测出的是数组单元的大小。表达式array n（其中n=0, 1, 2,）中，array扮演的是指针，故array n的结果是一个指针，它的类型是TYPE*，它指向的类型是TYPE，它指向数组第n号单元。故sizeof(array n)测出的是指针类型的大小。例十：int array[10]. int (*ptr)[10]. ptr=amp.array中，array代表数组本身。本节中提到了函数sizeof()，那么我来问一问，sizeof(指针名称)测出的究竟是指针自身类型的大小呢还是指针所指向的类型的大小？答案是前者。例如：int (*ptr)[10]. 则在32位程序中，有：
sizeof(int(*)[10])==4 sizeof(int [10])==40 sizeof(ptr)==4 实际上，sizeof(对象)测出的都是对象自身的类型的大小，而不是别的什么类型的大小。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