

C语言程序设计(第6章指针)6 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\\_A8\\_8B\\_c97\\_135152.htm](https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_AF_AD_E8_A8_80_E7_A8_8B_c97_135152.htm)

6.6 指针数组 前面介绍了指向不同类型变量的指针的定义和使用，我们可以让指针指向某类变量，并替代该变量在程序中使用；我们也可以让指针指向一维、二维数组或字符数组，来替代这些数组在程序中使用，给我们在编程时带来许多方便。下面我们定义一种特殊的数组，这类数组存放的全部是指针，分别用于指向某类的变量，以替代这些变量在程序中的使用，增加灵活性。指针数组定义形式：类型标识\*数组名[数组长度]例如：  
char \*str[4].由于[]比\*优先权高，所以首先是数组形式str[4]，然后才是与“\*”的结合。这样一来指针数组包含4个指针str[0]、str[1]、str[2]、str[3]，各自指向字符类型的变量。例如：int \*ptr[5].该指针数组包含5个指针ptr[0]、ptr[1]、ptr[2]、ptr[3]、ptr[4]，各自指向整型类型的变量。[例6-22] 针对指针数组的应用，我们分别用指针数组的各指针指向字符串数组、指向一维整型数组、指向二维整型数组

```
#include<stdio.h>#include<string.h>#include<stdlib.h>int main( ) { char *ptr1[4]={"china","chengdu","sichuang","chongqin"}. /* 指针数组ptr1的4个指针分别依此指向4个字符串*/ int i,*ptr2[3],a[3]={1,2,3},b[3][2]={1,2,3,4,5,6}. for (i=0;i<4;i++) printf("\n%s",ptr1[i]). /* 依此输出ptr1数组4个指针指向的4个字符串*/ printf("\n"). for(i=0;i<3;i++) ptr2[i]=&a[i]. /*将整型一维数组a的3个元素的地址传递给指针数组ptr2*/ for(i = 0. i<3. i++) printf("M", *ptr2[i]). printf("\n"). for(i = 0. i<3. i++) ptr2[i]=b[i]. /*传递二
```

维数组b的每行首地址给指针数组的4个指针\*/ for(i = 0. i  
printf("M M\n", \*ptr2[i], \*ptr2[i] 1).}程序中指针数组与所指对象的关系如图6 - 1 2所示。 ptr1指针数组中的4个指针分别指向4个字符串，如图6 - 11的a)所示，程序中依此输出；ptr2指针数组共有3个指针，若将整型一维数组a中各元素地址分别传递给指针数组的各指针，则ptr2[0]就指向a[0]；ptr2[1]就指向a[1]；ptr2[2]就指向a[2]。若将二维数组各行的首地址分别传递给指针数组的各指针，如图6 - 11 b)所示，这样一来， ptr2[0]就指向了b数组的第0行，该行有两个元素，其地址为ptr2[0]与ptr2[0] 1；相应指针数组第i个元素ptr2[i]指向的b数组的第i行两个元素地址分别为ptr2[i]与ptr[i] 1。运行程序

：chinachengdusichuangchongqin1 2 31 22 45 6 在处理二维字符数组时，我们可以把二维字符数组看成是由多个一维字符数组构成，也就是说看成是多个字符串构成的二维字符数组，或称为字符串数组。指针数组对于解决这类问题（当然也可以解决其它问题）提供了更加灵活方便的操作。有一点需要说明，若定义一个指针数组后，指针数组各元素的取值（即地址）要注意安全性。如定义指针数组：char \*ptr[3].我们说该数组包含三个指针，但指针的指向是不确定的，指针现在可能指向内存的任一地址。假定现在作语句：scanf("%s", ptr[i]), 则输入的字符串在内存的存放其地址由ptr[i]决定。除非给指针数组元素赋值安全的地址。[例6-23] 定义字符指针数组，包含5个数组元素。同时再定义一个二维字符数组其数组大小为5 \* 10，即5行10列，可存放5个字符串。若将各字符串的首地址传递给指针数组各元素，那么指针数组就成为名副其实的字符串数组。下面对各字符串进行按字典排序。在字

字符串的处理函数中，strcmp(str1, str2)函数就可以对两个字符串进行比较，函数的返回值> 0、= 0

```
#include#include#includemain( ){ char
*ptr1[4],str[4][20],temp[20]. /*定义指针数组、二维字符数组、
用于交换的一维字符数组*/ int i,j. for(i=0.i gets(str[i]). /* 输入4
个字符串*/ printf("\n"). for(i = 0. i ptr1[i] = str[i]. /* 将二维字符
数组各行的首地址传递给指针数组的各指针*/ printf("original
string:\n"). for(i = 0. i printf("%s\n", ptr1[i]). printf("ordinal
string:\n"). for(i = 0. i for(j=0. j if( strcmp( ptr1[j], ptr1[j + 1])>0) {
strcpy(temp,ptr1[j]). strcpy(ptr1[j], ptr1[j + 1]). strcpy(ptr1[j + 1],
temp). } for( i=0.i printf("%s\n" , ptr1[i]).}运行程序：jkjkdkddfs
fhfgkjkfgkfhkfgkgfkklgjjkdjdkoriginal
```

```
string:jkjkdkddfsfhfgkjkfgkfhkfgkgfkklgjjkdjdkordinal
string:fhfgkjkfgkfhkfgkgfkklgjjkdjdkjkjkdkddfs 程序中一定要注意
指针的正确使用。一旦将二维字符数组的各行首地址传递给
指针数组的各指针，则相当于给指针分配了安全可操作的地址，地址空间大小由二维字符数组来决定。当然也可由编译
系统为指针分配地址用于字符串的存放。[例6-24] 利
```

```
用malloc( )函数为指针分配存储空间，实现字符串的排序
。 #include#include#includemain( ){ char *ptr1[4],*temp. int i,j. for
(i=0.i { ptr1[i] = malloc(20). /*为指针数组各指针分配20字节的
存储空间*/ gets(ptr1[i]). } printf("\n"). printf("original string:\n").
for(i=0. i printf("%s\n",ptr1[i]). printf("ordinal string:\n"). for(i=0.i
for(j=0.j if(strcmp(ptr1[j],ptr1[j + 1])>0) { temp = ptr1[j]. /*利用指
向字符串的指针，进行指针地址的交换*/ ptr1[j] = ptr1[j + 1].
ptr1[j + 1] = temp. } for( i=0.i printf("%s\n" , ptr1[i]).}运行程序，其
```

结果与上述例6 - 2 3完全相同。 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](http://www.100test.com)