挑战30天C 入门极限：c_c 中的字符指针数组,指向指针的指针的含义 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/134/2021_2022__E6_8C_91_E6_88_9830_E5_A4_c97_134541.htm 就指向指针的指针,很早以前在说指针的时候说过,但后来发现很多人还是比较难以理解,这一次我们再次仔细说一说指向指针的指针! 先看下面的代码,注意看代码中的注解! //程序作者:管宁 //站点:www.cndev-lab.com //所有稿件均有版权,如要转载,请务必著名出处和作者 #include iostream>#include string>usingnamespacestd. voidprint_char(char* array[],intlen).//函数原形声明 voidmain(void) { //----------段1---------- char*a[]={"abc","cde","fgh"}.//字符指针数组 char* *b=a.//定义一个指向指针的指针,并赋予指针数组首地址所指向的第一个字符串的地址也就是abc\0字符串的首地址 cout//---------- //----------段2---------- char* test[]={"abc","cde","fgh"}.//注意这里是引号,表示是字符串,以后的地址每加1就是加4位(在32位系统上) intnum=sizeof(test)/sizeof(char*).//计算字符串个数 print_char(test,num). cin.get(). //---------- } voidprint_char(char* array[],intlen)//当调用的时候传递进来的不是数组,而是字符指针他每加1也就是加上sizeof(char*)的长度 { for(inti=0.i{ cout} } 下面我们来仔细说明一下字符指针

数组和指向指针的指针,段1中的程序是下面的样子:
char*a[]={"abc","cde","fgh"}. char* *b=a. coutchar *a[]定义了一个指针数组,注意不是char[], char[]是不能同时初始化为三个字符的,定义以后的a[]其实内部有三个内存位置,分别存储了abc\0,cde\0,fgh\0,三个字符串的起始地址,而这三个位置的内存地址却不是这三个字符串的起始地址,在这个例子中a[]是存储在栈空间内的,而三个字符串却是存储在静态内存空间内的const区域中的,接下去我们看到了char* *b=a.这里是定义了一个指向指针的指针,如果你写成char *b=a.那么是错误的,因为编译器会返回一个无法将char* *[3]转换给char *的错误,b=a的赋值,实际上是把a的首地址赋给了b,由于b是一个指向指针的指针,程序的输出cout结果是 abc cde fgh 可以看出每一次内存地址的 1操作事实上是一次加sizeof(char*)的操作,我们在32位的系统中sizeof(char*)的长度是4,所以每加1也就是 4,实际上是*a[]内部三个位置的 1,所以*(b 1)的结果自然就是cde了,我们这时候可能会问,为什么输出是cde而不是c一个呢?答案是这样的,在c 中,输出字符指针就是输出字符串,程序会自动在遇到\0后停止. 我们最后分析一下段2中的代码,段2中我们调用了print_array()这个函数,这个函数中形式参数是char *array[]和代码中的char *test[]一样,同为字符指针,当你把参数传递过来的时候,事实上不是把数组内容传递过来,test的首地址传递了进来,由于array是指针,所以在内存中它在栈区,具有变量一样的性质,可以为左值,所以我们输出写成了,cout到这里这两个非常重要的知识点我们都说完了,说归说,要想透彻理解希望读者多动手,多观察,熟能生巧!420){this.width=420}">内存结构示意图! 100Test 下载频道开通，各类考试题目直接下载。详细请