

总结函数的参数指针参数及const修饰的情况计算机等级考试  
PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_\\_E6\\_80\\_BB\\_E7\\_BB\\_93\\_E5\\_87\\_BD\\_E6\\_c97\\_645577.htm](https://www.100test.com/kao_ti2020/645/2021_2022__E6_80_BB_E7_BB_93_E5_87_BD_E6_c97_645577.htm)

```
void testn(int n) { n = 0. //正确，但外部n值未变，小心逻辑错误 } void testn(const int n) { n = 0. //编译错误，加const可以防止逻辑错误 } void testn(int* n) { n = new int(3). //正确，但外部指针n值未变，小心逻辑错误 *n = 0. //正确，n指向变量的值被改变 } void testn(int* const n) { n = new int(3). //编译错误，n是常量指针，加const可以防止逻辑错误 *n = 0. //正确，n指向变量值被改变 } void testn(const int* n) { n = new int(3). //正确，但外部指针n值未变，小心逻辑错误 *n = 0. //编译错误，加const可以防止指针指向的变量值被改变 } void testn(int** ppn) { *ppn = new int[3]. //正确，ppn指向的指针值被改变 **ppn = 6. //正确，ppn指向的最终变量值被改变 *ppn[0] = 7. //正确，ppn指向的最终变量值被改变 ppn = new int*[3]. //正确，但外部二级指针ppn值未变，小心逻辑错误 ppn[0] = new int(5). //正确，但外部二级指针ppn值未变，小心逻辑错误 } void testn(int** const ppn) { *ppn = new int[3]. //正确，二级指针ppn值被改变 **ppn = 6. //正确，ppn指向的最终变量值被改变 *ppn[0] = 7. //正确，ppn指向的最终变量值被改变 //ppn[0] = new int(5). //正确，同*ppn //ppn = new int*[3]. //编译错误，ppn是常量二级指针加const可以防止逻辑错误 } void testn( int* const* ppn) { /*ppn = new int[3]. //编译错误，ppn的值为常量指针，不能改变*ppn的值 **ppn = 6. //正确，二级指针指向的最终变量值被改变 *ppn[0] = 7. //正确，二级指针指向的最终变量值被改变
```

```
ppn = new int*[3]. //正确，但外部二级指针ppn值未变，小心  
逻辑错误 ppn[0] = new int(5). //编译错误，不能改变*ppn的值  
} void testn( int const** ppn) //同testn(const int ** ppn) { *ppn =  
new int[3]. //正确，二级指针ppn指向的指针值被改变 //**ppn  
= 6. //编译错误，不能改变**ppn的值 //**ppn[0] = 7. //编译错误  
， *ppn[0] = **ppn //ppn = new int*[3]. //编译错误，int ** 到  
const int ** 转换错误 }来源：www.examda.com //总结：函数参  
数为传值方式时，加const可以防止逻辑错误 //函数参数为传  
址方式时，加const可以防止参数数据被改变 //当参数为多级  
指针时，看const修饰对象可以看const与变量名之间的* //  
当const与变量之间没有*时，修饰的就是多级指针本身 //  
当const与变量之间有一个*时，修饰的是多级指针的值 //如此  
推算直到修饰最后的变量值，所以const放在变量类型前后效  
果相同 编辑特别推荐: 送给正在学习C 朋友的50条建议 C 笔试  
考前练习 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
```