

全国计算机等级考试四级上机编程修养(下) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022__E5_85_A8_E5_9B_BD_E8_AE_A1_E7_c98_138166.htm

6、if 语句对出错的处理 先看一段程序代码吧。if (ch >= ' 0 ' amp. ch /* 正常处理代码 */ }else{ /* 输出错误信息 */ printf("error\\n"). return (FALSE).} 这种结构很不好，特别是如果“正常处理代码”很长时，对于这种情况，最好不要用else。先判断错误，如：if (ch ' 9 '){ /* 输出错误信息 */ printf("error\\n"). return (FALSE).} /* 正常处理代码 */ 这样的结构，不是很清楚吗？突出了错误的条件，让别人在使用你的函数的时候，第一眼就能看到不合法的条件，于是就会更下意识的避免。

7、头文件中的#ifndef 千万不要忽略了头文件中的#ifndef，这是一个很关键的东西。比如你有两个C文件，这两个C文件都include了同一个头文件。而编译时，这两个C文件要一同编译成一个可运行文件，于是问题来了，大量的声明冲突。还是把头文件的内容都放在#ifndef和#endif中吧。不管你的头文件会不会被多个文件引用管你的头文件会不会被多个文件引用，你都要加上这个。一般格式是这样的：#ifndef #define #endif 在理论上来说可以是自由命名的，但每个头文件的这个“标识”都应该是唯一的。标识的命名规则一般是头文件名全大写，前后加下划线，并把文件名中的“.”也变成下划线，如：stdio.h #ifndef _STDIO_H_ #define _STDIO_H_ #endif (BTW：预编译有多很有用的功能。你会用预编译吗？) (BTW：预编译有多很有用的功能。你会用预编译吗？) 100Test 下载频道开通，各类考试题目直接下载。详细请访

