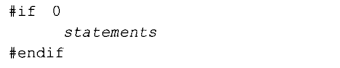
# C和指针

## C语言注释

注释时，符号/\*…..\*/之间不能在有其他的注释。且在C语言中，不能使用/\*…..\*/来注释代码，如果这段代码中原先就有注释存在，这样就会出问题。要从逻辑上删除这一段C代码，更好的方法就是使用#if指令。只要如下面使用：

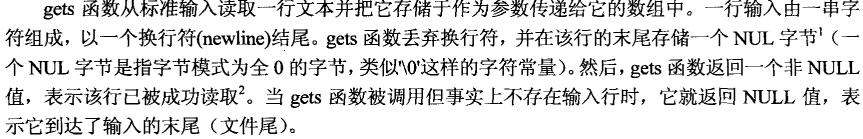


## 预处理指令

#include把包含的c文件导入到当前文件中，如果当前文件需要使用导入文件中的函数需要在当前文件前方引入函数原型，因此，可以在映入的文件中直接写上函数原型，那么在当前文件中也就不需要再写上函数院系了。

const表示这个函数不会修改函数调用者所传递的参数。

#define是另一种预处理，他将把名字替换为想要替换的东西。

C语言参数传值时，数组参数是传地址（引用）。而标量和常量则是传值。

C语言中并不存在字符串”String”,但在语言中，存在一项约定：字符串就是以NUL（全0字节）为结尾。字符串常量：“Hello”在内存中占据6个字节，按顺序分别是H、e、l、l、o和NUL。

&在C语言中被解释为取地址符号。&&是“逻辑与”操作。与C语言一样，&&操作连接的式子两端，如果左端为假，右边的表达式将不再进行求值，因此不管他是真是假，表达式总是假的。

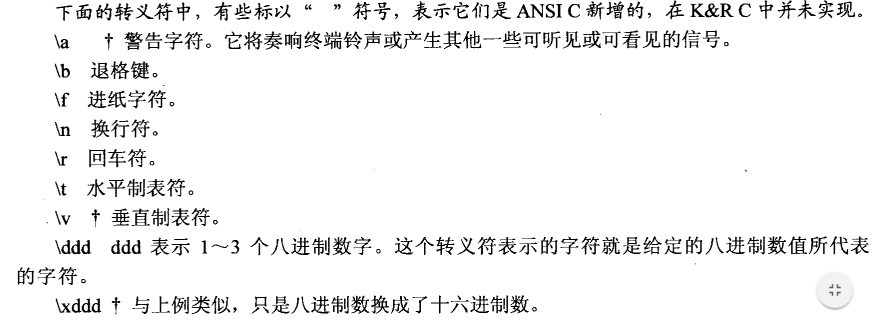
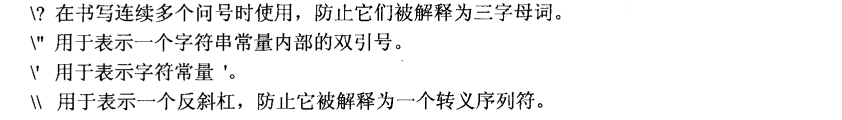
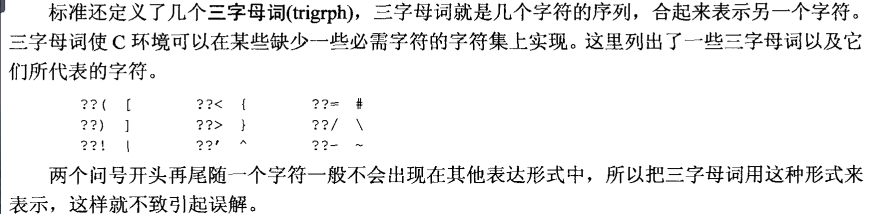
## 编译执行

编译过程也由几个阶段组成，首先是预处理器(preprocessor)处理。在这个阶段预处理器在源代码上执行一些文本操作。例如，用实际值代替#define指令定义的符号以及读入#include指令包含的文件内容。然后，源代码经过解析判断语句的意思。随后产生目标代码。之后优化器会对目标代码会进行进一步的处理，使他的效率更高。

## 文件名约定

C源代码以.c为扩展名保存。含有#include指令的C源代码称为头文件必须具有扩展名.h。

## 字符

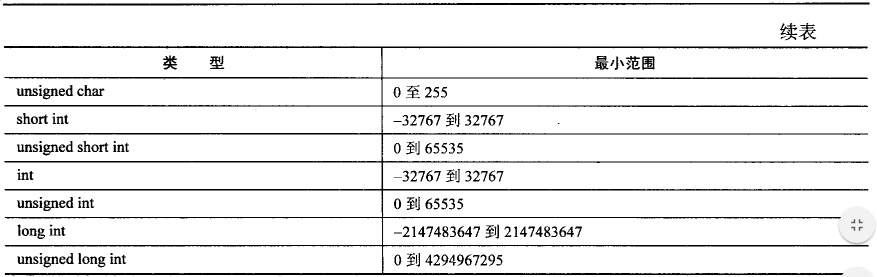
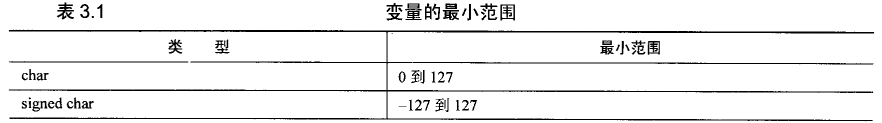


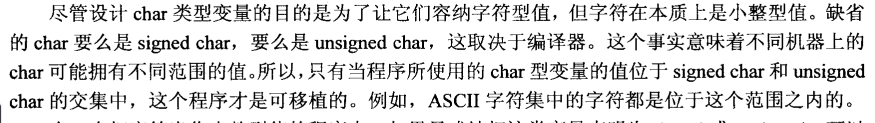
## 基本数据类型

四中基本数据累心：整型、浮点型、指针、和聚合类型(如数组和联合)

### 整形

整型家族包括字符、短整型、整型、和长整型，他们都有有符号和无符号数之分。而长整型至少应该和整型一样长，而整型至少应该和短整型一样长。

变量最小范围表。short int 至少16位(2 byte)，long int至少32位(4 byte)。Z至于缺省的int究竟是16位还是32位，或者是其他值则由编译器设计者决定。

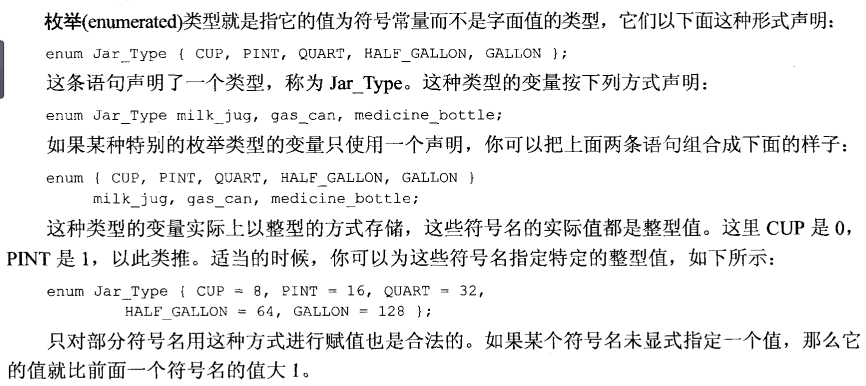
所以只有当程序所使用的char型变量位于Signed char和unsigned char的交集中，这个程序才能移植。例如，ASCII字符集中的字符都是位于这个范围之内的。

### 整型字面值

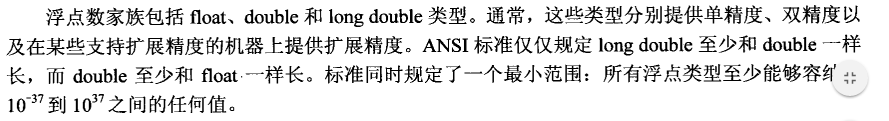
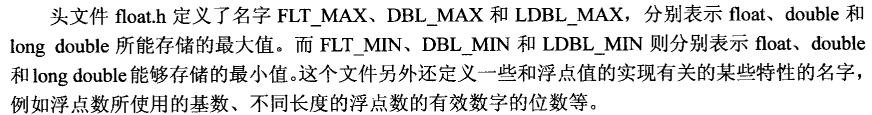
整型数字后面添加字符L或l，那么这个整数可以被解释为long整型值，字符U或u则用于把数值指定为Unsigned整型值。整型常量在缺省情况下，他是最短类型但能完整容纳这个值。整型也可以是八进制表示，只要数值前面以0开头。整数也可以是十六进制来表示，他以0x开头。

字符型常量也是允许的，例如’M’。最后，如果一个多字节字符常量前面有一个L，那么他就是宽字符常量，如L’x’、L’e^’。当运行时环境支持一个宽字节集时，就有可能使用他们。

## 枚举类型



## 浮点类型

浮点数字字面值在缺省的情况下都是double类型的，除非他的后面跟着L或l表示他是一个long double类型的值，或者跟一个F或f表示他是一个float类型的。

## 指针

指针常量：

字符串常量：字符串常量通常被书写方式是一对引号保卫一串字符，如“”“Hello”、“\aWarning\a”。

