

《现代电子技术》增刊

Turbo C 2.0

编程及应用速成

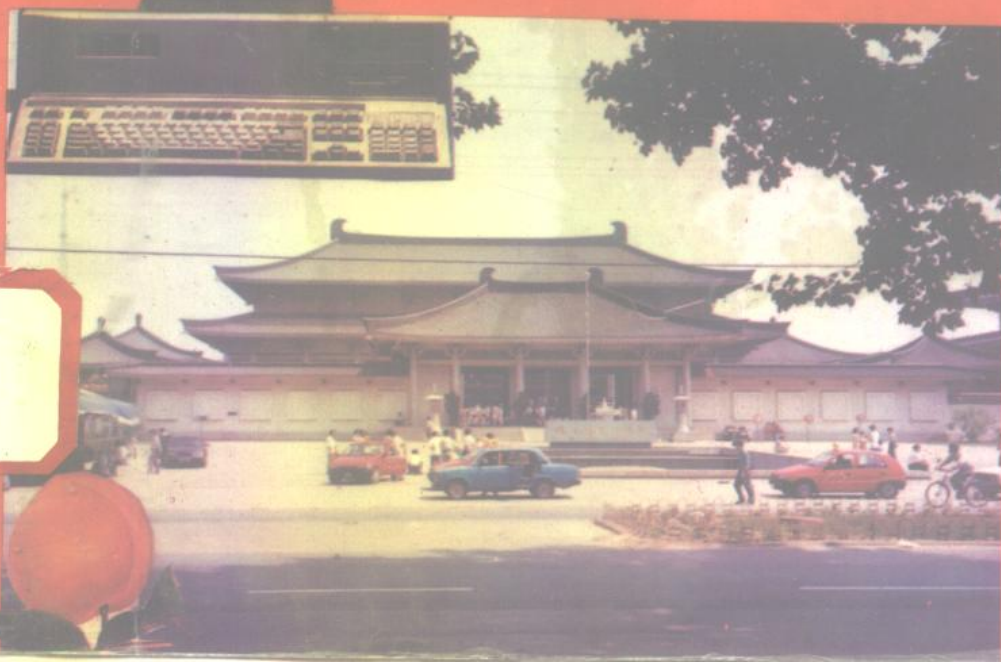
精彩示例：

文本窗口操作演示程序

弹出式窗口编辑器

动画模拟河内塔问题求解过程

平抛的动画演示



陕西电子杂志社

TP312

390113

J91

Turbo C 2.0 编程及应用速成

Turbo C 2.0 编程及应用速成

陕西电子杂志社

JS193/34 17

Turbo C 2.0 编程及应用速成

金 钊 编著

陕西电子杂志社出版发行

陕西杨陵科技印刷厂印刷

787×1092 毫米 1/16 开本

1995 年 10 月第 1 版 1995 年 10 月第 1 次印刷

印数:1—5000 册 22.75 印张 543 千字

国内统一刊号:CN61—1224/TN

定价:24.00 元

前 言

C 语言是当今世界上最流行的通用型程序设计语言,它是专门为开发系统软件而设计的,不但数据类型丰富,可移植性强,程序具有结构化特点,表达简洁,灵活多样,而且兼顾高级语言与汇编语言的优点,具有速度快,效率高等特点。

C 语言加入面向对象程序设计(OOB)的新型程序设计语言 C++ 与专门设计 Microsoft Windows 应用程序的 C++ for Windows 均是以 C 语言为基础的,并且 C 语言也是设计操作系统、编译系统、大型应用软件、游戏软件的主流语言,如何帮助读者迅速掌握 C 语言,进而进行实际应用也是作者编写此书的目的和愿望。

本书具有如下几个特点:

第一,IBM PC 及其兼容机上常用的 C 语言编译器很多,本书介绍的是其中最常见的,由美国 Borland 公司出品的高效 C 语言编译系统 Turbo C 2.0。它具有全新的交互式集成开发环境,集多项功能于一体;支持 IEEE 浮点标准,提供协处理器仿真程序;提供丰富的图形库函数,支持多种显示适配器;支持混合语言程序设计,可在 C 语言源程序中嵌入汇编语言;支持多种存贮模式,方便用户对存贮空间的有效管理。

第二,全书内容丰富,条理清晰,循序渐进。全书首先介绍绪论及 C 语言基础知识,然后依次介绍结构控制语句,函数,数组和指针,编译预处理,结构、联合与枚举和文件,最后介绍广大读者最感兴趣也是应用最为广泛的文本屏幕管理和图形绘制与屏幕管理。

第三,全书示例丰富。所有程序均使用 Turbo C 2.0 编译调试通过。程序采用标准缩进格式书写,美观大方,易于理解、阅读和修改。

每个示例均具有典型代表意义,深入浅出,个别程序中融入程序优化设计,一题多解等优秀程序设计思想和编程技巧。特别是全书重要章节末均附有应用实例,这些实例均密切联系本章所述内容,重点突出,具有实际应用价值。有些突出对典型问题的解决方法,如第三章的应用实例:逻辑推理问题求解、一元方程的近似解法,第五章的应用实例:排序等。有些突出对库函数的实际应用,如第四章的应用实例:按键选择,第七章的应用实例:结构在时间函数中的应用和系统中断调用等。还有些是较复杂的程序设计实例,如第八章的应用实例:通讯录管理程序,第九章的应用实例:文本窗口操作演示程序和弹出式窗口编辑器,还有第十章的应用实例:动画模拟河内塔问题求解过程和平抛的动画演示等。这些实例不但可以帮助读者理解该章所述内容,而且能使读者在理解的基础上达到举一反三、触类旁通的效果。

第四,为方便读者使用 Turbo C 2.0,本书书末附有 Turbo C 2.0 的安装, Turbo C 2.0 集成开发环境的使用和 Turbo C 2.0 库函数等使用和应用 Turbo C 2.0 最基础的内容。其中 Turbo C 2.0 集成开发环境的使用中不但介绍了集成开发环境中各项菜单的功能,并列举了所有编辑命令。书末还分类按字母顺序列出了 Turbo C 2.0 全部库函数,包括这些函数的功能、函数原型和函数用法,以方便读者查阅和使用。

本书可供各类院校师生作为学习 C 语言的教材使用,也可供广大计算机爱好者和有关

技术人员参考。

由于本人水平有限，书中不妥之处在所难免，敬请广大读者批评指正。

最后，要感谢父母对我的养育之恩，感谢王建国、刘子芳、杨康善、李人俊等老师对我的谆谆教导和精心培养，感谢所有关心、爱护、支持我的亲人、老师和朋友。

编者

一九九五年七月

于西安电子科技大学

目 录

第一章 绪论	(1)
1.1 C语言概述	(1)
1.2 Turbo C 概述	(2)
第二章 C语言基础知识	(3)
2.1 C语言程序基本结构	(3)
2.2 标识符	(4)
2.2.1 关键字	(5)
2.2.2 特定字	(5)
2.3 基本数据类型	(5)
2.3.1 字符型	(6)
2.3.2 整型	(7)
2.3.3 浮点型	(8)
2.3.4 无值型	(9)
2.3.5 变量初始化	(9)
2.4 运算符和表达式	(10)
2.4.1 算术运算符	(10)
2.4.2 关系运算符	(12)
2.4.3 逻辑运算符	(12)
2.4.4 位运算符	(13)
2.4.5 赋值运算符	(14)
2.4.6 条件运算符	(15)
2.4.7 逗号运算符	(16)
2.4.8 地址运算符	(16)
2.4.9 sizeof()运算符	(16)
2.4.10 其它运算符	(17)
2.4.11 优先级和结合性	(18)
2.5 数据类型转换	(19)
2.5.1 混合运算中的类型转换	(19)
2.5.2 强制类型转换	(20)
2.6 标准输入输出函数	(20)
2.6.1 格式化输入输出函数	(20)
2.6.2 非格式化输入输出函数	(32)

2.7 常用数学函数.....	(35)
2.8 字符处理函数.....	(37)
2.9 简单程序设计.....	(39)
第三章 结构控制语句	(41)
3.1 程序的三种基本结构.....	(41)
3.2 选择结构控制语句.....	(43)
3.2.1 if 语句.....	(43)
3.2.2 if—else 语句.....	(44)
3.2.3 嵌套 if 语句.....	(45)
3.2.4 switch 语句.....	(45)
3.3 循环语句.....	(53)
3.3.1 for 语句.....	(53)
3.3.2 while 语句.....	(56)
3.3.3 do—while 语句.....	(59)
3.3.4 循环的嵌套.....	(61)
3.3.5 break 语句.....	(63)
3.3.6 continue 语句.....	(65)
3.4 标号和 goto 语句.....	(65)
3.5 应用实例：逻辑推理问题求解.....	(67)
3.5.1 好事是谁做的.....	(67)
3.5.2 对竞赛名次的预测.....	(68)
3.5.3 破案.....	(70)
3.6 应用实例：一元方程的近似解法.....	(71)
3.6.1 对分法.....	(71)
3.6.2 迭代法.....	(73)
3.6.3 牛顿法.....	(74)
第四章 函数	(76)
4.1 函数定义.....	(76)
4.1.1 函数定义的一般形式.....	(76)
4.1.2 函数返回值与 return 语句.....	(78)
4.1.3 函数说明与函数原型.....	(80)
4.2 函数调用.....	(81)
4.2.1 函数调用的一般形式.....	(81)
4.2.2 函数的多级调用.....	(86)
4.2.3 函数的递归调用.....	(89)
4.3 局部变量和全局变量.....	(94)
4.3.1 局部变量.....	(94)

4.3.2 全局变量.....	(95)
4.4 存贮类型及作用域规则.....	(96)
4.4.1 自动变量.....	(97)
4.4.2 外部变量和外部函数.....	(97)
4.4.3 静态变量和静态函数.....	(99)
4.4.4 寄存器变量.....	(100)
4.5 应用实例：按键选择.....	(101)
第五章 数组和指针.....	(109)
5.1 数组.....	(109)
5.1.1 一维数组.....	(109)
5.1.2 多维数组.....	(113)
5.1.3 字符串与字符数组.....	(119)
5.2 指针.....	(121)
5.2.1 指针的使用.....	(121)
5.2.2 指针运算符.....	(121)
5.2.3 指针运算.....	(123)
5.2.4 无类型指针.....	(124)
5.3 指针和数组.....	(125)
5.3.1 指针和一维数组.....	(125)
5.3.2 指针和二维数组.....	(127)
5.3.3 指向数组的指针.....	(129)
5.3.4 指针数组.....	(129)
5.3.5 指向指针的指针.....	(131)
5.4 引用调用.....	(132)
5.4.1 指针变量作为函数参数.....	(133)
5.4.2 数组名作为函数参数.....	(134)
5.5 命令行参数.....	(137)
5.6 指针函数.....	(140)
5.7 字符串处理函数.....	(141)
5.8 函数指针.....	(143)
5.9 应用实例：排序.....	(147)
5.9.1 冒泡排序.....	(147)
5.9.2 选择排序.....	(148)
5.9.3 线性插入排序.....	(149)
5.9.4 对半插入排序.....	(150)
5.9.5 快速排序.....	(150)

第六章 编译预处理	(153)
6.1 宏定义	(153)
6.2 文件包含	(156)
6.3 条件编译	(157)
6.4 预处理操作符 # 和 ##	(159)
6.5 预定义宏	(160)
第七章 结构、联合与枚举	(162)
7.1 结构	(162)
7.1.1 结构的定义与使用	(162)
7.1.2 结构数组	(165)
7.1.3 结构与指针	(168)
7.1.4 结构与函数	(169)
7.1.5 位域	(174)
7.2 动态分配函数	(177)
7.3 引用自身的结构	(180)
7.4 联合	(181)
7.5 枚举	(185)
7.6 类型定义	(187)
7.7 应用实例：结构在时间函数中的应用	(188)
7.8 应用实例：系统中断调用	(191)
第八章 文件	(193)
8.1 流和文件系统	(193)
8.1.1 缓冲 I/O 与非缓冲 I/O	(193)
8.1.2 流	(193)
8.1.3 文件结构	(194)
8.1.4 预定义流	(195)
8.2 缓冲文件系统	(196)
8.2.1 文件的打开和关闭	(196)
8.2.2 字符输入输出函数	(198)
8.2.3 格式化输入输出函数	(200)
8.2.4 数据块读写函数	(202)
8.2.5 定位函数	(204)
8.2.6 错误检测函数	(205)
8.3 非缓冲文件系统	(206)
8.3.1 文件柄	(207)
8.3.2 文件的建立、打开和关闭	(207)

8.3.3 文件的读写	(209)
8.3.4 定位函数 lseek()	(211)
8.4 应用实例：通讯录管理程序	(212)
第九章 文本屏幕管理	(221)
9.1 设置文本显示方式	(221)
9.2 文本窗口的定义及操作	(224)
9.3 窗口内的输入输出	(226)
9.4 文本屏幕块操作	(229)
9.5 应用实例：文本窗口操作演示程序	(232)
9.6 应用实例：弹出式窗口编辑器	(239)
第十章 图形绘制与屏幕管理	(259)
10.1 图形系统控制函数	(259)
10.1.1 图形模式初始化	(259)
10.1.2 图形模式与文本模式的转换	(263)
10.2 颜色控制函数	(263)
10.3 绘图函数	(265)
10.3.1 基本图形函数	(265)
10.3.2 绘图方式设置函数	(267)
10.4 图形模式下的文本输出	(270)
10.4.1 文本输出函数	(270)
10.4.2 文本字体设置函数	(271)
10.5 屏幕管理	(274)
10.5.1 图块操作函数	(274)
10.5.2 视口管理函数	(276)
10.5.3 多页屏幕管理函数	(277)
10.6 应用实例：动画模拟河内塔问题求解过程	(278)
10.7 应用实例：平抛的动画演示	(282)
附录 A Turbo C 2.0 的安装	(285)
附录 B Turbo C 2.0 集成开发环境的使用	(289)
B.1 集成环境的组成	(289)
B.2 集成菜单的使用	(291)
B.3 编辑命令	(300)
附录 C Turbo C 2.0 库函数	(303)
C.1 分类函数	(303)

C.2 目录函数	(304)
C.3 转换函数	(306)
C.4 检测函数	(307)
C.5 输入输出函数	(308)
C.6 接口函数	(318)
C.7 串和内存操作函数	(324)
C.8 数学函数	(328)
C.9 存贮分配函数	(334)
C.10 进程控制函数	(336)
C.11 标准函数	(337)
C.12 信号函数	(338)
C.13 时间和日期函数	(339)
C.14 变量参数表函数	(340)
C.15 文本与图形处理函数	(341)
C.16 其它函数	(351)
附录 D 扩展的键盘扫描码	(353)

第一章 绪 论

1.1 C 语言概述

C 语言与 UNIX 操作系统是密不可分的。最初的 C 语言只是为描述和实现 UNIX 操作系统而设计的。随着 UNIX 操作系统的广泛流行, C 语言也迅速得到推广, 现已成为世界上最流行的通用型程序设计语言。

归纳起来, C 语言具有如下特点:

一、C 是中级语言

以往的系统软件通常用汇编语言编写, 不但移植性差, 而且可靠性低, 开发极不方便。C 语言是专门为开发系统软件而设计的, 它兼顾高级语言与汇编语言的优点, 使系统软件的开发更为方便、快捷。

在 C 语言中, 可以象汇编语言一样对位、字节和内存单元直接进行操作。C 语言编译程序简单而紧凑, 生成目标代码质量高, 程序执行效率高, 一般只比汇编程序生成的目标代码效率低 10%~20%。

二、C 语言数据结构丰富

C 语言具有丰富的数据结构。如基本数据类型: 字符型、整型、浮点型、双精度浮点型和无值型; C 语言本身提供的构造数据类型: 数组类型、指针类型、结构类型、联合类型与枚举类型。C 语言还提供了类型定义语句, 利用它, 用户可使用已有的数据类型方便地构造其它数据类型, 如堆栈、树、链表等。

C 语言的各种数据类型中, 指针最能体现 C 语言的各种特点。利用指针, 可以方便灵活地操作各种复杂数据类型的变量, 而且可以对函数进行某些特殊的操作。C 语言高度灵活的表达能力在一定程度上来自于巧妙而恰当的使用指针。

三、C 是结构化语言

结构化语言的显著特点是代码及数据的分隔化, 即程序的各个部分除必要的信息交换外彼此独立。这种结构化方式可使程序层次清晰, 便于使用、维护和调试。C 语言是以函数形式提供给用户的, 并且每一个函数都是独立的, 可以单独编译, 便于程序分块加工和调试。C 语言具有多种循环语句、条件语句控制程序流程, 从而使程序完全结构化。

四、C 语言具有预处理功能

编译预处理是 C 语言区别其它高级语言的特征之一。C 语言的预处理在通常的编译

(包括词法与语法分析、代码生成、优化等)之前,它是 C 语言编译程序的一部分。常用的预处理功能包括宏定义、文件包含和条件编译。

C 语言的预处理功能为程序员提供了有效的工具,使得程序员利用预处理功能编制的程序易于实现,便于调试和移植到不同的计算机上去。

五、C 语言可移植性能好

C 语言是和 UNIX 操作系统相结合而发展起来的。随着 UNIX 操作系统在各种机器上实现的同时,C 语言也迅速得到推广。并且现在许多其它操作系统也都配置有 C 语言的编译程序,这些编译器虽然在不同操作系统下实现,但却是以同一标准——ANSI C 作为基础的,这使得 C 语言具有良好的可移植性能,许多程序可以从一个编译器拿到另一个编译器下直接编译通过。

PC 机上常用的 C 语言编译程序有 Turbo C、Microsoft C、Quick C 等,本书中的程序使用 Turbo C 进行编译,大部分程序也可使用 Microsoft C 和 Quick C 编译,个别程序稍经修改也可在 Microsoft C 和 Quick C 下运行。

1.2 Turbo C 概述

Turbo C 2.0 是美国 Borland 公司开发的用于 IBM PC 及其兼容机上的一个高效的 C 语言编译系统,它为用户提供了 400 多个标准库函数和 6 个系统实用程序。其全新的交互式集成开发环境,支持系统提供的大部分功能。其中包括高效的全屏编辑器,可用来书写各种文本文件;提供友善的用户接口,方便的下拉式菜单,将程序编辑、编译、连接、运行一体化;支持工程文件,便于程序模块化开发和维护;具有单步执行、断点设置、监视和表达式计算等功能,方便程序调试;强大的及时帮助功能。

同时,Turbo C 系统还具有如下特点:

支持 IEEE 浮点标准,当没有 8087 或 80287 协处理器时,可使用系统提供的仿真 8087 或 80287 协处理器实用程序进行高速的浮点运算。

提供丰富的图形库函数,支持 CGA、EGA、VGA、Hercules、3270PC、AT&T400 和 IBM8514 等多种显示适配器,并可连接用户的图形驱动程序和矢量字体。

可直接与汇编语言进行连接,支持混合语言程序设计。也可采用嵌入式汇编,在 C 语言源程序中直接嵌入汇编程序。

支持极小、小、中、紧凑、大和特大等六种存贮模式,方便用户对存贮空间的有效管理。

第二章 C 语言基础知识

2.1 C 语言程序基本结构

了解 C 语言程序的基本组成部分,对 C 语言编程有一个感性认识,是进行 C 语言程序设计的基础,也便于读者对后继课程的学习。

下面分析一个简单的 C 语言程序,该程序调用一个求和函数 sum() 计算 15+20+25 的和。

```
1: #include "stdio.h"    /* 包含头文件 stdio.h */
2:
3: int sum(int a,int b,int c);    /* 自定义函数 sum() 的原型说明 */
4:
5: int main(void)    /* 定义主函数 */
6: {
7:     int a,b,c,d;    /* 定义 a,b,c,d 为整型变量 */
8:     a=15,b=20,c=25;
9:     d=sum(a,b,c);    /* 调用函数 sum() 计算 a+b+c 的和 */
10:    printf("%d+%d+%d=%d\n",a,b,c,d);    /* 输出计算结果 */
11:    return 0;
12: }
13:
14: int sum(int a,int b,int c)    /* 定义函数 sum() */
15: {
16:     return(a+b+c);    /* 返回函数计算结果 */
17: }
```

C 语言中,有一类函数是系统本身提供的,称为库函数,在使用这些库函数时,必须在程序最前面包含定义这些库函数的头文件。该程序中,使用了格式化输出库函数 printf(), 该函数原型在 stdio.h 中说明,所以程序第 1 行用预处理命令 #include 包含此头文件。

程序第 3 行使用函数原型说明一个用户自定义的函数 sum(), 该函数包含三个参数,均为 int 型(整型),函数返回值也是 int 型。

程序第 5 行开始定义主函数 main(), 它没用参数(void 表示没有参数),返回值为 int 型,程序第 6 行到第 12 行为 main() 的函数体,包含在一对花括号“{”和“}”中,其中第 9 行调用函数 sum() 求 a+b+c 的值并赋于 d。

程序第 14 行到 17 行定义函数 sum(), 它有三个参数 a、b、c, 返回 a+b+c 的值。

注意程序第 3 行与第 14 行非常相似,仅差一个分号,但起的作用截然不同。

C 语言程序还有以下几个特点：

1. C 语言区分大小写，例如变量 A 与 a 不同。程序通常用小写字母书写，只有在个别情况下才使用大写字母。
2. 函数是 C 语言程序的基本单位，一个 C 语言源程序由一个或多个函数组成，其中至少包含一个主函数 main()。C 语言程序总是从 main() 函数开始执行的。
3. 函数体必须用花括号 “{” 和 “}” 括起来。
4. 分号是语句和数据定义的终止符。
5. 注释包含在 “/*” 和 “*/” 之间。对 C 语言源程序进行编译时，程序注释将被忽略。

C 语言程序书写格式灵活，可将几条语句写在一行上，举例如下：

```
#include "stdio.h"

int sum(int a,int b,int c);

int main(void) { int a,b,c,d;a=15,b=20,c=25;d=sum(a,b,c);printf("%d+%d+%d=%d\n",a,b,c,d);return 1;}

int sum(int a,int b,int c) { return(a+b+c);}
```

这样写在语法上并没用错误，但阅读起来很不方便，同时也使得程序的层次不明确。建议读者在编制 C 语言程序时，不要将多条语句写在同一行上，遇到嵌套语句向后缩进，必要时给程序加上注释，这样可以使程序结构清晰、易于阅读、维护和修改。

一个较完整的 C 语言程序应包括：预处理、类型定义、用户自定义函数原型说明、全局变量说明、主函数定义、若干子函数定义等几个主要部分。

2.2 标识符

C 语言中，用来标识变量、常量、数据及函数名字的有效字符序列称为标识符。标识符的定义必须符合下列语法规则。

- (1) 标识符只能由字母、数字和下划线组成，并且第一个字符必须是字母或下划线。
- (2) 在缺省状态下，Turbo C 只能识别标识符的前 32 个字符，如果两个标识符的前 32 个字符相同，Turbo C 就认为是同一个标识符。有效字符数目可以通过集成环境的 O|C|Source|Identifier Length 选项修改。详见附录 B.2。

- (3) Turbo C 中，大小写字母表示不同含义，构成不同的标识符。

根据定义，下面的标识符是合法的：

```
NAME
PIC _1
count
Inter _80486
_key
```

而下列标识符是非法的：

```
student.da 存在非字母数字字符
```

5dollar 数字不能作为标识符的第一个字符

标识符的取名最好根据它所代表的含义取其英文或汉语拼音，或其字母缩写、字头来作标识，以便于阅读和检查。

Turbo C 中，存在两类特殊的标识符——关键字和特定字，下面将分别讨论。

2.2.1 关键字

关键字又称为保留字，是已被 Turbo C 使用，具有特定的含义，用来说明变量的类型、作为语句定义符号等用途的标识符，是 C 语言的特定部分，不得赋予其它含义。

Turbo C 关键字由两部分组成，一部分是由 ANSI 标准推荐的关键字，另一部分是 Turbo C 扩展的关键字。

由 ANSI 标准推荐的 32 个关键字如下：

auto	break	case	char	const
contiune	default	do	double	else
enum	extern	float	for	goto
if	int	long	register	return
short	signed	sizeof	static	struct
switch	typedef	union	unsigned	void
volatile	while			

属 Turbo C 扩展的 11 个关键字如下：

asm	_cs	_ds	_es	_ss
cdecl	far	near	huge	interrupt
pascal				

2.2.2 特定字

特定字是具有特定含义的标识符。这些标识符虽不是关键字，但具有特定含义，不能作为一般标识符使用，这些特定字是：

define	include	undef	ifdef	ifndef
endif	line			

它们主要用在 C 语言的预处理程序中，使用方法详见第六章。

2.3 基本数据类型

C 语言中所有变量在使用前必须加以说明，确定该变量的数据类型。C 语言有五种基本

数据类型: 字符型、整型、浮点型、双精度型和无值型, 用于说明这些变量的关键字是: char、int、float、double 和 void。

除 void 类型外, 其它四种基本数据类型前面都可以加类型修饰符, 用于改变基本类型的取值范围, 以便更适合各种情况的需要。下面列出常用的各种类型修饰符。

signed	带符号数
unsigned	无符号数
long	长的
short	短的

2.3.1 字符型

一、字符变量的说明

定义字符型变量的关键字是 char, 加上不同的修饰符, 通常定义的字符型变量有 char、signed char、unsigned char 三种类型。它们各自在机器中所占的位数和取值范围见下表。

类 型	位 数	范 围
char	8	-128~127
unsigned char	8	0~255
signed char	8	-128~127

字符型变量的定义形式如下:

```
char c1;           /* 定义 c1 为字符型变量 */
unsigned char c2;  /* 定义 c2 为无符号型字符变量 */
```

二、字符型常量表示

字符常量是用单引号括起来的一个字符, 如 'x'、'O'、'!' 等都是字符常量。由于字符常量在实际存过程中, 存放的并不是字符本身, 而是字符的 ASCII 码, 所以字符常量也可以用 ASCII 码直接表示, 如十进制数 65 表示字符 'A', 十进制数 97 表示的是小写字母 'a', 用这种方法可以得到一些特殊字符, 如十进制数 27 表示 ESC, 十进制数 13 表示回车 (CTRL-M) 等。

除了上述两种表示方法外, 还可用反斜杠(\)开头的字符序列, 即转义序列表示字符常量。如 '\n' 表示换行。

反斜杠与八进制或十六进制数一起可用于表示该数值在 ASCII 码表中所对应的字符, 可用 '\ ' 后跟 1 至 3 位八进制数表示, 如 '\033' 表示 ESC, '\101' 表示字母 'A', 也可用 '\x' 或 '\X' 后跟 1 至 2 位十六进制数表示字符常量, 如 '\X3F' 表示 '?', '\x0d' 表示回车。

下表列出 Turbo C 中的转义序列。