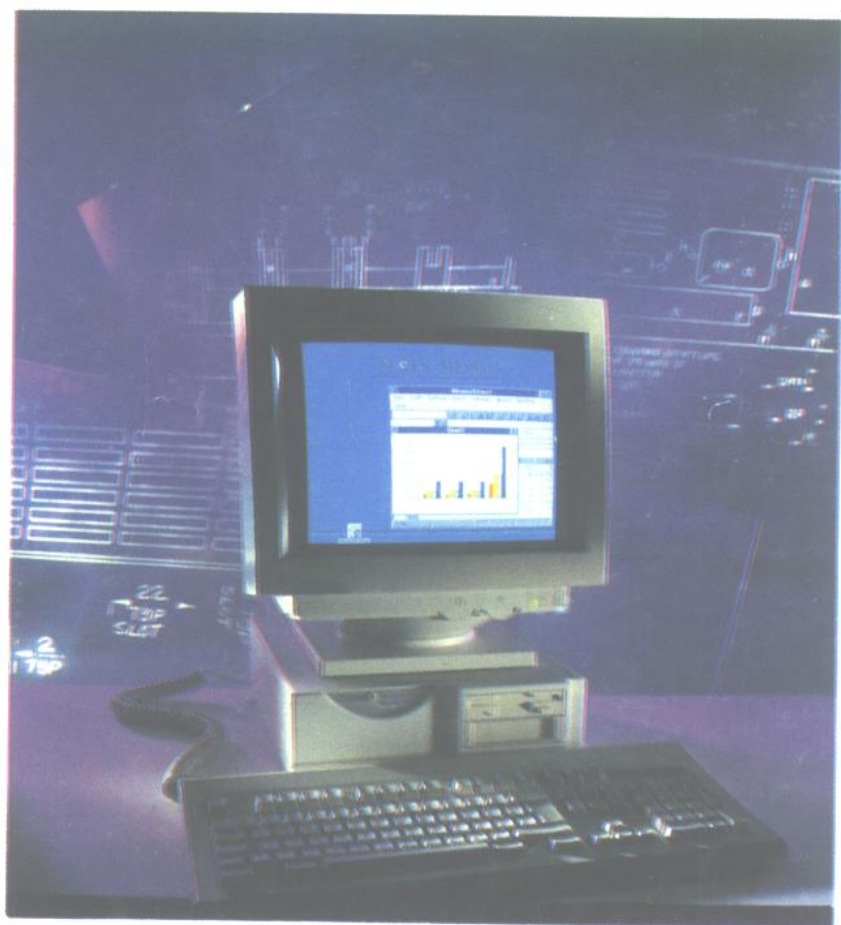


●计算机基础教育系列教材●

C语言基础教程

C语言 基础教程

周大庆 主编



中南工业大学出版社

中南

312
2011

社

C 语言基础教程

周大庆 主编

黄大明 徐红云 骆嘉伟 编著
陈启买 周大庆

中南工业大学出版社

C 语言基础教程

周大庆 主编

责任编辑:谢贵良

*

中南工业大学出版社出版发行

长沙交通学院印刷厂印装

湖南省新华书店经销

*

开本:787×1092 1/16 印张:15 字数:372 千字

1996 年 3 月第 1 版 1996 年 3 月第 1 次印刷

印数:0001—5000

*

ISBN 7-81020-838-1/TP·062

定价:15.00 元

本书如有印装质量问题,请直接与承印厂家联系解决

内 容 简 介

本书以 87 ANSI C(美国国家标准 C 语言)为基础,以 PC 机上的 Turbo C 2.0 版本为背景,全面、系统地介绍了 C 语言的基本内容和程序设计方法。每章附有习题,供读者训练与自测学习效果。

本书可作为大、中专学校和计算机培训班的教材,也可供科技人员、管理人员和其他自学者参考。

前 言

C语言是目前国内外广泛流行的一种计算机程序设计语言,它具有简洁、高效和接近汇编语言的特点,既适合编写系统程序,又适合编写应用程序。它自问世以来,以其独特的魅力吸引了广大计算机用户,学习和使用C语言已成为当今计算机程序人员的一种新时尚。

本书以 87 ANSI C(美国国家标准C语言)为基础,以 PC 机上的 Turbo C 2.0 版本为背景,全面、系统地介绍了C语言的基本概念和语法规则(数据类型、数据运算、各种语句和程序结构)以及用C语言进行程序设计的方法。

近几年国内出版了不少关于C语言方面的专著与教材,但它们对基础教育(中专生、大专文科生等)的读者不是很适合。本书起点低,用大量示例说明C语言技术,同时兼顾提高。

本书内容新颖,叙述简明,由浅入深,通俗易懂。书中提供了大量的程序示例和习题,所有程序均在 Turbo C 2.0 环境下调试通过。为便于查阅,书末还提供了库函数总表和出错信息表等四个附录。

本书可作为大、中专学校和计算机培训班的教材,也可供科技人员、管理人员和其他自学者参考。

本书第1、2、3章由黄大明执笔,第4、5、6章由徐红云执笔,第7、8、9章由骆嘉伟执笔,第10、14章由陈启买执笔,第11、12、13、15章由周大庆执笔,全书由周大庆主编并最后修改定稿。

限于编者的经验和水平,书中难免存在不足之处,欢迎读者批评指正。

编 者

1996年2月

目 录

第 1 章 C 语言概述	(1)
1.1 C 语言的演进轨迹	(1)
1.2 独具特色的 C 语言	(3)
1.3 C 程序的结构	(4)
1.4 Turbo C 的上机步骤	(5)
习题	(7)
第 2 章 基本数据类型	(8)
2.1 C 语言的字符集	(8)
2.2 标识符和关键字	(8)
2.3 基本数据类型	(9)
习题	(14)
第 3 章 运算符与表达式	(15)
3.1 算术运算符.....	(15)
3.2 强制类型转换运算符.....	(17)
3.3 基本赋值运算符和基本赋值表达式.....	(17)
3.4 复合赋值运算符和表达式.....	(20)
3.5 sizeof 运算符	(20)
3.6 逗号运算符和逗号表达式.....	(21)
3.7 关系运算符和关系表达式.....	(22)
3.8 逻辑运算符和逻辑表达式.....	(22)
3.9 其它运算符.....	(23)
习题	(25)
第 4 章 顺序控制结构	(27)
4.1 C 语言的基本语句	(27)
4.2 数据的输入和输出.....	(29)
4.3 顺序结构程序示例.....	(38)
习题	(39)
第 5 章 选择控制结构	(42)
5.1 if 语句	(42)
5.2 switch 语句	(46)
5.3 选择结构程序示例.....	(51)
习题	(54)

第 6 章 重复控制结构	(56)
6.1 for 语句	(56)
6.2 while 语句	(60)
6.3 do-while 语句	(61)
6.4 break 语句和 continue 语句	(62)
6.5 goto 语句和标号	(64)
6.6 重复结构程序示例	(65)
习题	(70)
第 7 章 数组	(73)
7.1 数组概述	(73)
7.2 一维数组	(74)
7.3 二维数组	(79)
7.4 字符数组	(83)
习题	(86)
第 8 章 函数	(89)
8.1 函数的定义与调用	(89)
8.2 函数的返回值	(92)
8.3 函数的参数	(95)
8.4 函数与数组	(98)
8.5 递归函数	(101)
习题	(104)
第 9 章 存储类别与作用域	(106)
9.1 局部变量和全局变量	(106)
9.2 存储类别	(110)
习题	(115)
第 10 章 编译预处理	(117)
10.1 文件包含	(117)
10.2 宏	(118)
10.3 条件编译	(123)
习题	(126)
第 11 章 指针	(127)
11.1 指针的概念	(127)
11.2 指针的运算	(129)
11.3 指针与数组	(130)
11.4 指针与函数	(132)
11.5 动态存储分配	(137)
11.6 指针数组	(137)
11.7 指针的指针	(140)
习题	(140)

第 12 章 字符串	(144)
12.1 字符串的处理方法	(144)
12.2 字符串函数	(146)
12.3 字符串数组	(150)
12.4 命令行参数	(152)
习题	(153)
第 13 章 结构体与联合体	(155)
13.1 结构体的概念	(155)
13.2 结构体数组	(159)
13.3 指向结构体的指针	(160)
13.4 嵌套结构体	(163)
13.5 链表	(164)
13.6 联合体	(166)
13.7 类型定义	(167)
13.8 枚举类型	(168)
习题	(169)
第 14 章 位运算与位段	(172)
14.1 位运算	(172)
14.2 位段	(177)
习题	(181)
第 15 章 文件的输入与输出	(182)
15.1 文件的概念	(182)
15.2 缓冲型文件系统	(182)
15.3 文件定位与随机存取	(191)
15.4 非缓冲型文件系统	(195)
15.5 标准设备文件	(197)
习题	(197)
附录 I ASCII 字符表	(199)
附录 II Turbo C 编辑器命令	(200)
附录 III Turbo C 库函数总表	(202)
附录 IV Turbo C 编译错误信息表	(225)

第 1 章 C 语言概述

本章简要介绍 C 语言的历史和特点、C 程序的格式和结构特点，以及程序的上机调试步骤，以期勾画出 C 语言的整体轮廓，使读者尽早对 C 语言的全貌有一个粗线条的认识。

1.1 C 语言的演进轨迹

在一片赞誉声中，K. Thompson 和 D. M. Ritchie 因写作 UNIX 操作系统和 C 语言的巨大成功，获得了 1982 年度的 Electromics 最高成就奖。该项大奖自设立以来还是第一次授予在计算机软件领域里作出杰出贡献的科学家。继而他们又摘取了 1983 年度计算机科学最高荣誉奖——图灵奖。毫无疑问，UNIX 和 C 巨大的商品价值和显要的学术地位，已成为计算机科学前进历程中的一座传世的丰碑。

自 1969 年开始，美国贝尔实验室的 K. Thompson 和 D. M. Ritchie 用了不到二年的时间，以汇编语言为工具，研制出了 UNIX 操作系统，并成功地在 PDP-7 上实现。由于汇编语言过分地依赖于计算机硬件且因机型不同而异，其可移植性、可读性都很差。因此，一个很自然的想法是：能否找到一种语言工具，既具有汇编语言的某些功能（例如对内存地址的操作，位操作等），又具有高级语言的功能？

在当时的计算机程序语言家族中，由英国剑桥的 Martin Richard 开发的 BCPL (Basic Combined Programming Language) 语言因具有如下几个特点而引起了人们的注意：

- 它是结构化程序设计语言。
- 能够直接处理与机器本身数据类型相近的数据。
- 具有与内存地址对应的指针处理方式。

它是在 CPL (Combined Programming Language) 语言基础上开发出来的，作为记述语言使用的一种程序语言，是一种单一数据类型语言。

美国贝尔实验室的 K. Thompson 继承和发展了 BCPL 语言的优点，摒弃其缺点，从中脱胎出 B 语言，“B”这一名字取自 BCPL 的第一个字母。

K. Thompson 用 B 语言重写了 UNIX 操作系统，并在 DEC 公司研制的 PDP-7 机上实现。继而又在 PDP-11/20 机上成功地用 B 语言实现了 UNIX 操作系统。这一关键性的迈步是令人鼓舞的，它使人们见到了用高级语言写作系统软件的希望，尽管 B 语言还存在着如下的主要不足之处：

- PDP-11 是按字节编址的（现在的微机也是如此），而 B 是面向字的，这样就妨碍了对单个字节进行存取的能力。

- B 语言仍然和 BCPL 语言一样，数据无类型，它只有一种机器字类型。因此，在描述客观世界的事物时会遇到许多困难。

- B 编译程序产生的是解释执行代码，运行速度较慢。

为了克服上述不足，D. M. Ritchie 作了不懈的努力，获得了丰硕的回报。1972 年他成功地开发出 C 语言。1973 年，他又和 K. Thompson 合作，用 C 把 UNIX 的核心程序及其庞大的外层软件，包括系统命令、实用程序、库以及各种应用软件重写了一遍，这就是著名的 UNIX V5(第五版)——它一举奠定了 UNIX 系统的基础。以后 UNIX V6、V7、System III 和 System V 都是在 V5 的基础上发展、扩充的。

今天，当 UNIX 操作系统风靡世界，并在各种机型(包括微机)上安了家的时候，人们无不称赞 C 语言的巨大功绩，并为 C 语言所具有的简洁、实用、代码质量高，表达能力强，特别是可移植性好等优良性能所惊叹！

“锥处囊中，必然脱颖而出”。C 语言，曾经是与 UNIX 操作系统相依相伴而生息繁衍的孪生兄弟，因其自身所具有的优良素质，现在已远远地超出了 UNIX 系统的范围，坚定执着地走着自己辉煌的生命之路。

各种 C 语言编译程序及软件环境应运而生，产生了众多功能各异的 C 语言支持系统和应用系统。并在各种机型(包括微、小、中、大型机)上安营扎寨。如果说 C 语言在这一阶段作为通用程序语言还只是雏型，那么，1978 年，以 UNIX 第 7 版中的 C 编译程序为基础，B. W. Kernighan 和 D. M. Ritchie 合著的传世之作《The C Programming language》(简称 K&R C)则标志着 C 语言作为独立于 PDD-11 和 UNIX 的一种通用程序语言正在走向成熟。

K&R C 可称为 C 语言的“圣经”，它成为后来各种广泛使用的 C 语言版本的基础，因而被称为标准 C。此后，在陆续出现的各种不同的 C 语言版本中，较出名的有：C86、BDS C、INSTANT C、Lattice C、Microsoft C、Quick C 等。

1983 年，美国国家标准局成立的一个专门委员会，针对标准 C 问世后出现的各种不同版本对 C 语言的若干发展加以补充和修改，公布了一个 C 语言美国国家标准草案，并于 1988 年获准实施，称之为 ANSI C，并使之成为 C 语言的“新约圣经”。此后，C 语言各种编译系统都向此标准靠拢。

同年，K&R 完全按照 ANSI C 标准修改了他们的经典著作《The C Programming Language》。

ANSI C 比原标准 C 有了很大的发展，引入了许多新的结构，类型和功能，最重要的是：

- 在数据表示、数据结构、数据运算上的变化，如说明和定义函数变量的描述、异常错误处理、类型说明与扩展、运算符增删等。

- 定义了标准库，规定了诸如访问操作系统、格式化 I/O、存储器分配、字符串处理、数字计算等函数，并形成了若干对应的标题文件(.h 文件)，通过对这些标题文件的修改达到与主系统的密切配合和程序兼容。

近年来，随着集成软件的发展，出现了一些 C 语言的集成开发环境。这种环境是集屏幕编辑、语言编译、模块连接、动态调试和程序运行于一体，并配之以窗口、图形、高效率交互方式(如鼠标)等支持的一个功能强大的 C 语言工具环境，一旦进入此环境就可以从头到尾完成程序的编制、调试和运行全部过程。

Borland 国际公司的 Turbo C2.0 就是这类集成软件的杰出代表之一。它模拟 C 语言在 UNIX 上的运行环境，参照 ANSI C 语言标准，把 C 移植在 DOS 上运行，同时扩展了许多功能。Turbo C 的具体优点如下：

- 有优良的用户界面。

- 编辑、编译、连接、运行一气呵成。
- 快速。
- 可移植性好，兼容性高。

本书将以 ANSI C 为标准，以 Turbo C 为工作平台，全面、系统、深入浅出地介绍 C 语言的基本内容和程序设计方法。

1.2 独具特色的 C 语言

C 语言一经问世，便显示出强大的生命力，令人们趋之若鹜，被认为是目前世界上用途最广泛的软件工程语言。确实，C 语言除具有其它一些语言所具备的素质外，另有其独特的过“人”之处：

- 语言简洁、精炼、紧凑。

C 语言只用一个字母命名，很容易使人联想到它的简洁和精炼。确实，学过任何一门高级语言的人，只要稍作比较，就会感受到这一点。C 语言一共只有 32 个关键字、9 种控制语句，压缩了一切不必要的成分。这一点可在本章下一节所举例题中去分析体会。

- 所占内存空间少。

C 语言编译程序小而精，大约只有 12K 左右，这是因为 C 语言没有内部函数。例如：C 没有提供 I/O 设施，也没有并行操作、同步或协同程序等复杂控制，而是提供了大量而有效的库函数来实现 I/O、字符串处理、存储器分配、数学运算等操作。而这些库函数本身并不属于 C 语言源程序，它们被分门别类地存放在不同的程序库中，需要时再加以引用。

- 生成的代码质量高，程序运行速度快。

完成同样的任务，你会发现用 C 语言编写的程序的运行速度快于以其它高级语言（如 BASIC、PASCAL）所编写的程序。

- 语言表达能力强。

以上几条，究其原因如下：

C 语言当初的设计目的就是用来编写系统软件的，因此，它具有系统层次的操作能力，如输入输出操作，对硬件端口，对字符、数字的地址进行逐位操作甚至对寄存器、中断向量等进行直接操作的功能，因而生成的代码质量高。其生成代码的效率仅比用汇编语言编写的代码低 10%~20%。但它又具有高级语言的属性：描述问题编程迅速、工作量小、可读性好、易于调试和修改。因此，C 语言在编写系统软件方面大有取代汇编语言之势。

由于 C 语言兼有“系统描述语言”和“通用程序设计语言”的双重特性，人们称之为“中级语言”。

- 是一种面向过程的语言。

因为 C 的函数与一般的程序设计语言不同之处是既能当函数值传递，又可当做子程序使用，就连主程序也被视为是一个函数。

- 是一种面向程序的语言。

它可用包含方式引入程序库中的函数，也可让用户自行定义函数以扩大程序库，即根据自己的行业特点创建自己个人的程序库，从而大大地提高了程序生产效率。这是一种很有弹性的语言。

- 是一种结构化语言。

C 语言具有支持顺序、选择、重复三种基本结构的控制语句，三种基本控制结构通过串联、并联和嵌套可以组成复杂的程序。

- 可移植性好。

一旦程序编制完成，不论在哪一种机器上都不需要更改或只需稍作改动即可运行。诚然，几乎所有的高级语言都有很强的可移植性，但 C 语言作为一种系统描述语言能具有这一特性，就显得更为难能可贵了！

1.3 C 程序的结构

下面给出一个完整的 C 语言源程序，读者不必顾及程序中每个语句的功能。仔细观察这个程序，你会发觉它对于了解 C 语言源程序的结构特点是大有裨益的。

例 1.1

[程序]

```
/* This is the first example */
#include <stdio.h>
main()
{
    int n;
    long factor();
    long y, z;
    y=factor(5);          /* 调用求阶乘的子程序 */
    z=factor(7)+factor(15);
    printf("5! = %ld\n", y);
    printf("7! +15! = %ld\n", z);
}
long factor(x)
int x;
{
    long y;
    for (y=1; x>0; x--)
        y=y*x;
    return(y);
}
```

[运行结果]

5! =120

7! +15! =2004315056

本程序第一行是 `/* ... */`。这是 C 程序中的注释语句，在对 C 源程序进行编译时，注释语句并不产生目标代码。换句话说：注释语句并不是 C 语言程序的一部分，因而在程序中是可有可无的。但作为一个好的程序员，应养成在程序中必要的地方加上注释语句的良好习惯，以增强程序的可读性。注释语句可出现在程序的任何地方。

本程序第二行的形式为 `"#include < * * *.h>"`，称为预处理语句。其中“#”表示这是由预处理程序处理的行，include 表示本程序包含后面括号里指定文件的内容，stdio.h 是文件

名, 该文件提供了引用标准 I/O 库所需要的说明信息。

本章 1.2 节已经述及, C 语言没有内部函数。也就是说: 标准库函数并不是 C 语言的一部分, 库中的所有函数只是支持 C 语言形成一个系统编程环境。当程序中需要使用这些功能性操作时, 可象上面程序的第二行一样, 将 C 函数库中的相应函数包含进来。必须指出的是: 库的组成和大小可以随 C 编译系统的不同而不同, 但附属于 C 的标准库是基本一致的。Turbo C 函数库中大约提供了三百个标准函数, 详见附录 I。

本程序余下的部分是两个函数, 一个是主函数 `main()`, 另一个是名为 `factor` 的求阶乘的函数。在主函数中三次调用 `factor` 函数, 求出 $5!$ 和 $7! + 15!$ 的值, 然后输出结果。

一个 C 语言源程序中可以包含一个或者多个函数, 但必须有且只能有一个主函数 (`main()`)。 `main()` 函数可放在程序的任意位置。C 程序执行时, 总是先从 `main()` 函数开始, `main()` 函数执行完毕后, 整个程序也就执行完了。

C 语言源程序中各函数是相互独立的, 它们在程序中的地位平等, 互不包容, 也就是说: 函数中不能再定义函数。

作为一个训练有素的 C 程序员, 应养成这样一种良好的程序设计风格: 用多个函数构成一个大程序, 而每个函数是独立的, 且具有完成某一特定的操作的功能。同时, 也可将这些独立的函数视为一个个“零件”。在需要时, 用这些“零件”装配成各种“机器”(程序), 而不必每次重新去“铸造”这些“零件”。

C 语言函数的一般格式如下:

函数名(参数表)

参数说明

{

变量说明部分

执行语句部分

}

C 程序的书写格式如下:

(1) C 程序的书写格式自由, 一行中可以有多个语句, 一个语句也可以分成几行书写。每个语句(包括说明语句)最后必须有一个分号。分号是 C 语句必要的组成部分。为了增强程序的可读性, 同时使程序书写规范, 建议最好一行写一个语句。

(2) 在 C 语言中大写字母和小写字母是有区别的。习惯上除常量名用大写字母以外, 其余的一般都用小写字母书写。

(3) 若程序中出现了语句嵌套的情况, 建议采用向后缩进的书写格式。

1.4 Turbo C 的上机步骤

Turbo C 是一个集编辑器、编译器、连接器于一体, 并配上文件管理程序及检错公共程序而构成的一种集成式软件, 它采用菜单驱动与交互式处理, 用户界面非常友好。读者可在 Turbo C 的菜单指引下, 无师自通, 得心应手地使用 Turbo C 这一强有力的工具。

Turbo C 提供了两种不同的开发程序的方法, 其中一种是传统的命令行方式, 这种方式必须要有 `TCC.EXE` 文件存在当前目录下。

首先, 用户可用任何一种文本编辑程序(例如 PE、WORDSTAR、EDLIN 或 EDIT 等)编

辑自己的 C 语言源程序。程序编完存盘后，在操作系统提示符下，直接键入 TCC 及源程序文件名即可对它进行编译、连接，生成可执行程序文件(. EXE 文件)，然后在操作系统提示符下直接运行该程序即可得到结果。

Turbo C 第二种产生程序的方法是集成开发环境，在这个环境下可以完成从编辑、编译、连接到执行并观察执行结果的全过程。

如果你的机器上安装了 Turbo C 系统，在 TC 子目录下，你只要键入 TC 并按回车键即可进入 Turbo C 集成开发环境，此时屏幕内容如图 1.1 所示。

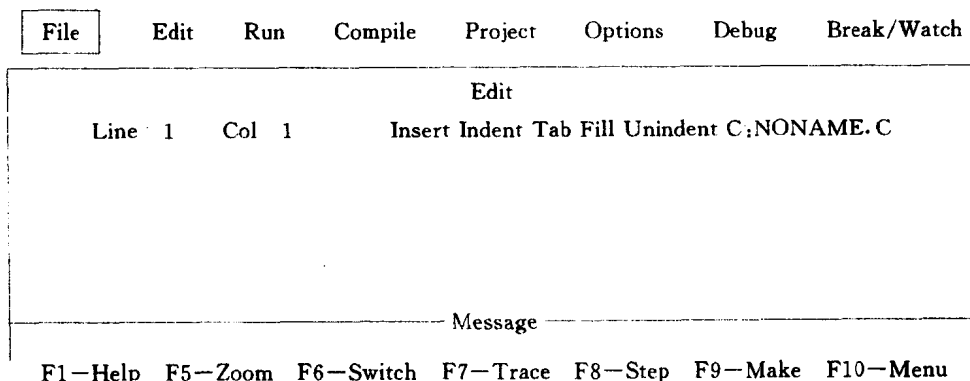


图 1.1 Turbo C 集成开发环境的屏幕

屏幕的最上面一行是主菜单，其中列出了集成开发环境所提供的主要功能。任何时候同时按下 Alt 键和主菜单选择项的首字母(F、E、R、C、O 等)即可进入相应的菜单选择项。屏幕的最下面一行是热键提示行，其中列出了某些常用操作所对应的热键组合。

屏幕中部有两个窗口，上面的是编辑窗口，下面的是消息窗口。编辑窗口顶端的状态行给出了光标所在的行和列、插入/重写状态及正在编辑的文件名等信息。消息窗口用来在编译和连接程序时显示警告或出错信息。

在编辑窗口，按键盘上通常的可打印字符键，都被看作是文本的输入字符，将其显示在光标位置，然后光标向右移动一格。

键盘上的 4 个箭头键↑、↓、←、→按照箭头指示的方向，可把光标移到编辑窗口的任意位置。

Home 和 End 键分别使光标跳到本行的行首和行尾。PgUp 键使编辑窗口显示上一页的文本；PgDn 键使编辑窗口显示下一页的文本。

Ctrl-Home 和 Ctrl-End 键分别使光标跳到屏幕的顶行和底行。Ctrl-PgUp 和 Ctrl-PgDn 键分别使光标跳到正在编辑的文件的第一行和最后一行。

Backspace 键删除光标左边的一个字符，Del 键删除光标所在处的一个字符。Ctrl-Y 删除光标所在的行。

进入编辑窗口之后如果从未按过 Ins 键，则保持缺省的插入状态。按一次 Ins 键则关闭插入状态，变成重写状态。在这种状态下按回车键只使光标回到本行左端，而不能插入新的一行。再按一次 Ins 键则恢复插入状态。

按 Ctrl-K-B 键设置块的开始，按 Ctrl-K-K 键设置块的结尾。按 Ctrl-K-C 键将块复制到当前光标位置，按 Ctrl-K-V 键将块移动到当前光标位置，按 Ctrl-K-Y 键删除标记块。

在编辑窗口常用的热键组合有下面几个：

F3	从磁盘将文件装入编辑窗口
Ctrl-F9	编译、连接并执行当前编辑窗口的程序
Alt-F5	切换到用户屏幕
F6	编辑窗口与消息窗口的切换键
F2	存储编辑窗口的文件
Alt-X	退出 Turbo C 环境，回到 DOS 环境

习 题

- 1.1 将 C 语言与 BASIC 语言或其它高级语言的特性作比较，阐述 C 语言的特点。
- 1.2 为什么说 C 语言是一种较“低级”的高级语言，与汇编语言有什么区别？
- 1.3 C 语言与标准库有什么关系？C 语言系统的库包含了些什么？
- 1.4 什么是标准 C？什么是 ANSI C？两者有什么不同？
- 1.5 上机运行本章的例题，熟悉 Turbo C 系统的上机操作步骤。

第 2 章 基本数据类型

C 语言的字符集、标识符和数据类型是 C 语言程序设计的基本元素，如同建房屋所用的砖块一样。读者通过本章的学习，将为进一步学好 C 语言打下必要的基础。

2.1 C 语言的字符集

C 语言字符集由下列几部分组成：

- 数字：0, 1, 2, 3, ……，9。
- 英文字母：A, B, C, ……，Z; a, b, c, ……，z。
- 下划线：“_”，这个字符与一个英文字母的地位相当。在构成标识符等语法成分时，以下划线“_”打头在 C 语言中是合法的，而在其它语言中通常是不允许的。
- 一些特殊用途的 ASCII 字符，如 &、*、/、{、[、,、; 等等。

2.2 标识符和关键字

1. 标识符

标识符用来为变量、类型和函数命名，名称必须以字母或下划线开头，后面只能跟若干个字母、下划线或者数字。例如 total, _below, sum, _file, _FILE, Lotus1_2_3 等都是合法的标识符；D. M. Ritchie, \$ 5, # 23, 3D 等是不合法的标识符。必须注意，_file 和 _FILE 是两个不同的标识符，因为在 C 语言中大小写字母是有区别的。

标识符的长度限制无统一规定，随系统不同而异。有些系统只取 8 个字符，如果标识符的长度超过了 8 个字符，则只有前 8 个字符有效。

读者在定义标识符时，最好是根据被标识的对象取名，真正起到“标识”的作用，达到“见名知义”的效果。

2. 关键字

关键字是语言本身所规定的用字，在程序中具有特殊的意义，不能作为一般标识符使用。C 语言共有 32 个关键字。

用作定义和说明数据类型的关键字有：

auto	char	const	double	enum
extern	float	int	long	register
short	signed	sizeof	static	struct
typedef	union	unsigned	void	volatile

用作流程控制语句的关键字有：

break	case	continue	default	do
else	for	goto	if	return
switch	while			

2.3 基本数据类型

C语言的数据类型是非常丰富多彩的，它构成了C语言的“脊梁”。可以这么说，谁如果掌握了C语言的全部数据类型，再加上有程序设计的基础，那么，他就大体上掌握了C语言。

C语言的数据类型大致可分为两大类：

- 基本类型

基本类型包括：整型、字符型、浮点型和双精度浮点型。

- 构造类型

构造类型包括：数组类型、结构体类型、联合体类型、枚举类型和指针类型。

构造类型是C语言的精华之所在，也是学习C语言的重点和难点。读者在学习这些内容时应认真、投入，务必仔细领会，熟练掌握。本书将在以后的章节中对这些内容另立专题讨论。本章将重点介绍基本类型。

2.3.1 各种类型的常量

常量是在程序运行过程中，其值不能被改变的量。

C语言中有各种各样的常量，包括整型常量、浮点型常量、字符常量和字符串常量等。

1. 整型常量

整型常量有下面三种形式：

(1) 十进制整数。由数字0~9组成，可以带正负号，如10, 1234, -32768, 0等。

整型常量在机器上占一个字的空间，由于各种机器的字长不一样，所以整数类型所表示数的范围也有所不同。以16位计算机为例，如果是带正、负号的整数，其值在-32768~32767之间，如果不带符号的整数，其值在0~65535之间。

一个整数常量，如在后面加一个字母l或L，则认为是长整型(long int型)常量。例如15L, 15l, 100L, 0l等。长整型常量的取值范围在-2147483648~2147483647之间。

(2) 八进制整数。由数字0打头，后随一串八进制数字表示。例如037表示八进制数的37，即十进制数的31。

(3) 十六进制整数。以0x或0X开头，后随一串十六进制数字表示。例如0X1D表示十六进制数的1D，即10进制数的29。

2. 浮点型常量

C语言中的浮点数类型如同其它语言中的实数类型。浮点型常量有两种表示形式：

(1) 十进制小数形式。由数字和小数点组成，且必须有小数点。如1.0, 0.0, 3, 14159265, -0.123等。

(2) 指数形式。由整数部分、小数点、小数部分、e(或E)及带正负号的指数组成(正号可省略)。如0.314e+1, 0.314e1, 1.5e-5, 314e-2分别代表3.14, 3.14, 1.5×10^{-5} , 3.14。

浮点型常量在计算机内部都是以双精度存放的，但把浮点型常量赋给一个变量时，其数