

天津市高等学校计算机课程规划教材

高福成 编著

C程序设计 教程



天津大学出版社

TP312

G/61

天津市高等学校计算机课程规划教材

C 程序设计教程

高福成 编著
边奠英 主审

天津大学出版社

内 容 提 要

本书以应用为目的,以提高程序设计能力为目标,以程序设计方法学为依据,系统介绍了 C 语言及其程序设计技术。全书共十一章,包括 C 语言的初步介绍、基本数据类型、数据运算、程序控制结构、数组、函数、指针、复合数据类型、文件、编译预处理和大规模程序的开发技术等,并配以大量的应用实例和多种类型的习题。本书技能训练和能力培养并重,普及与提高兼顾,可以满足不同读者的需要。

本书内容深入浅出,循序渐进,方便自学,可作为高等学校本科和研究生及计算机等级考试培训班教材,也可供软件开发人员参考。

JS197/03

天津市高等学校计算机课程规划教材

C 程序设计教程

高福成 编著

边莫英 主审

*

天津大学出版社出版

(天津大学内)

邮编:300072

河北省邮电印刷厂印刷

新华书店天津发行所发行

*

开本:787×1092 毫米¹/₁₆ 印张:18¹/₄ 字数:456 千

1998 年 1 月第一版 1998 年 1 月第一次印刷

印数:1—5000

ISBN 7-5618-1011-3

TP·109 定价:20.00 元

天津市教委高等学校计算机课程规划教材编写委员会

主任委员	孙衍广	天津市教委
副主任委员	张庆生	天津市教委高教处
	边奠英	天津大学
	刘璟	南开大学
委员	(以姓氏笔画为序)	
	于长云	天津理工学院
	方大寿	天津轻工业学院
	朱守仁	天津财经学院
	曲建民	天津师范大学
	刘荫铭	天津职业大学
	李兰友	天津纺织工学院
	杨秀雯	天津大学出版社
	张金铎	天津大学出版社
	林成春	天津广播电视大学
	高福成	天津商业学院
	韩 劼	天津医科大学
	程国毅	天津市教委高教处
秘书	唐安娜	天津市教委高教处
主编	边奠英	
副主编	张庆生	刘璟

天津市高等学校计算机课程规划教材

序 言

本世纪80年代以来,计算机科学有了惊人的发展,计算机应用也已深入到社会生产、生活的各个领域。江泽民同志在党的第十五次代表大会上进一步阐明了“科学技术是第一生产力,科技进步是经济发展的决定性因素”的理论。这意味着,作为一个国家生产力高低和科学技术现代化程度重要标志之一的计算机应用水平,今后在我国必须有极大的提高。

我国高等院校历来十分重视培养学生具备计算机基础知识和应用能力的工作。尤其对大量的非计算机专业学生来说,这种培养工作尤为重要。因为我们的社会主义现代化需要大批的计算机应用人才。在现在以至将来,没有计算机知识、不能使用计算机,是很难满足工作的要求的。因此,培养工作必须同时面向非计算机专业的学生。“工欲善其事,必先利其器。”只有使这些学生成为既掌握本专业知识和技能、又精通计算机应用的综合性人才,才能在我国社会主义现代化建设中发挥更大的作用。

天津市是我国的一个重要工业城市,有几十所高等院校,每年有数万学生需要学习计算机课程。然而,多年来我市缺少自己的计算机教材。教师选择教材基本上是各行其是,既不统一,也不规范。这不能不影响我市计算机教学整体质量的提高,不能不影响学生计算机应用水平的培养。因此,我们认为,编写一套天津市高等学校适用的计算机教材已势在必行。

编写教材是一件十分严肃的工作,是一项大的工程。为此,我们委托天津市教委计算机基础课程指导委员会承办此事。他们聘请了天津市各高校有丰富教学经验的教师组成写作班子。这些老师们认真负责、一丝不苟。他们反复讨论写作大纲,并仔细推敲书稿中的每一个字和每一句话。这套计算机教材基本上反映了近几年计算机科学与技术的最新发展,符合国家教委对计算机课程的要求。我们殷切地希望广大学生、教师和专家提出宝贵意见,以便再版时修改和补充。

在这套教材的出版过程中,各方人士给予了大力支持和帮助,特别是天津大学出版社始终积极配合。在此,我们一并表示衷心的感谢。

天津市教育委员会

1997.11.

前 言

C语言是计算机程序设计语言中的小字辈,但卓越的性能使它风靡全世界,成为最受欢迎的语言之一。当今流行的面向对象语言 C++ 及 Internet 网上的 JAVA 语言就来源于 C,要学习 C++ 或 JAVA,先要掌握 C。

目前许多高等院校都已或正准备开设 C 语言课程。参加计算机等级考试的应试者中,选择 C 语言的人数不断增加。从这几年全国和天津市计算机等级考试的情况看,许多应试者缺少 C 语言的正规训练,C 语言的精华和要害部分常常掌握不好。为了适应形势的需要,为了尽快培养一批熟练掌握 C 语言及其程序设计技术的人才,天津市教委组织和指导了本书的编写工作。

在计算机日益普及、计算机技术日新月异、新型软件大量涌现的今天,程序设计仍然是每一个当代大学生的基本功,是计算机素质教育的重要一环。尤其对那些希望用计算机解决本专业领域实际问题的有识之士,更需要加强程序设计训练。

学习计算机语言的唯一目的是应用,而应用要通过程序设计来体现。程序设计发展到今天,已经由技艺推进到科学,有了自己的一套基本原理和方法。进行程序设计,需要很强的逻辑思维能力,是一种极富创造性的智力劳动。对初学者而言,最使他们望而生畏的,往往也正是这一点。可以这样认为,语言是一种技能,程序设计是一门科学。因此,任何计算机语言及其程序设计的基本特点就是理论性和实践性并重,教学上应强调科学训练与技能培养并存。基于这一认识,本书以应用为目的,以提高程序设计能力为目标,以程序设计方法学为依据,系统介绍了 C 语言及其程序设计技术,把程序设计作为科学来讲授,把语言作为技能来培养,让读者在大量的程序设计实践中自然而然地熟悉和掌握语言。

本书从选材上注意了以下几点。

(1) 立足基点掌握制高点 基点是会用语言来进行程序设计,学会一大批基本的程序设计单元;制高点是掌握程序设计的基本思维方法和一定的技巧。因此,本书对语言本身的问题点到为止,不深加追究,但对程序设计则根据实用和适用的原则,精选大量的例题加以阐述。这些例题不是围绕语言展开,而是服从程序设计需要的实际题目。

(2) 突出重点分解难点 本书的重点章节是第四、六、七、八章。书中对此进行了较大篇幅的阐述。对难点问题,如指针、数据在函数间的传递和函数库的使用等,则把它们分散在多个章节内讨论。

(3) 为了使读者能及时考查自己的学习效果,每一章都安排了大量的习题,包括选择题、填空题、编程题和程序调试题共四种题型,可满足读者参加等级考试、发挥程序设计的创造性以及上机实践的需要。

根据 C 语言自身的特点,本书从内容上做了如下安排:第一章是 C 语言初步介绍,尽管简单,但它是学习 C 语言的纲;第二、五、八章从易到难介绍 C 语言的各种数据结构;第三章专门介绍 C 语言的计算功能;第四章集中介绍 C 语言的控制结构和程序设计的基本思维方法,体

现结构化程序设计的特点;第七、九两章介绍函数和文件,这是模块化程序设计的需要;第六章介绍指针,这是 C 语言的精华和特色,也是学习的难点;最后两章介绍了 C 语言环境中的特殊而较为深入的问题,这些问题基本不影响本书对初学者的要求,但对较高层次的学员来说又是必要的资料,可以作为自学之用;附录部分给出了 C 语言的运算符、Turbo C 函数库和编译错误信息,便于读者查阅。

本书的对象既可以是初学者,也可以是有一定语言基础的学员。除了最后两章可作为自学内容外,其余九章对于初学者,约需讲授 50 学时;对于具有其他计算机语言基础的学员,约需讲授 36 学时。另外需要 36 学时上机实践。

边莫英教授审阅了全书,天津市教委还聘请了李芸和孙慧两位同志审阅了原稿。他们提出许多宝贵的建议,使作者获益匪浅。

在本书的编写和出版过程中,得到天津市教委和天津商学院领导的大力支持,天津大学出版社的领导和编辑也付出了大量的辛勤劳动,在此一并表示由衷的感谢。

尽管天津市教委要求出精品、出成果,作者也力图不负重托,但本人终究才疏学浅、力不从心,书中错误难免,请广大读者不吝指正。

编著者 谨识

1997.8.20

目 录

第一章 C 语言初步介绍	(1)
1.1 C 语言的起源和发展	(1)
1.2 C 语言的特点	(2)
1.3 C 程序的基本结构	(3)
1.4 简单的 C 程序设计	(8)
1.5 C 程序的编译与运行	(12)
1.6 Turbo C 2.0 集成环境	(13)
习题一	(21)
第二章 基本数据类型、常量和变量	(23)
2.1 基本数据类型	(23)
2.2 常量及其类型	(25)
2.3 变量及其类型	(29)
2.4 符号常量	(34)
2.5 不同类型数据的输入输出	(36)
习题二	(44)
第三章 数据运算	(49)
3.1 算术运算	(49)
3.2 赋值运算	(53)
3.3 逗号运算(顺序运算)	(56)
3.4 关系运算和逻辑运算	(58)
3.5 测试数据长度运算符 sizeof	(60)
3.6 位运算	(61)
3.7 数学函数	(66)
习题三	(67)
第四章 程序控制结构	(72)
4.1 if 语句	(73)
4.2 switch 语句和 break 语句	(80)
4.3 循环语句	(84)
4.4 转移控制语句(break、exit()、continue、goto)	(90)
习题四	(94)
第五章 数组	(100)
5.1 数组的定义和初始化	(100)
5.2 数组的赋值	(104)
5.3 数组的输入和输出	(105)

5.4 数组的应用	(110)
习题五	(115)
第六章 指针	(120)
6.1 地址和指针	(120)
6.2 指针的运算	(128)
6.3 指针与一维数组	(132)
6.4 指针和字符串	(133)
6.5 指针和二维数组	(137)
6.6 二级指针	(143)
6.7 指针与内存动态分配	(146)
习题六	(150)
第七章 函数	(156)
7.1 C程序的模块结构	(156)
7.2 C函数的定义	(156)
7.3 C函数的调用	(159)
7.4 函数间数据传递方式之一——哑实结合	(161)
7.5 函数间数据传递方式之二——函数返回值	(173)
7.6 函数间数据传递方式之三——全局变量	(174)
7.7 函数的嵌套调用和递归调用	(175)
7.8 main()函数的参数和返回值	(179)
习题七	(181)
第八章 复合数据类型	(186)
8.1 结构类型和结构	(186)
8.2 结构数组	(192)
8.3 指向结构和结构数组的指针	(193)
8.4 结构在函数间的传递	(196)
8.5 递归结构	(201)
8.6 位段结构	(204)
8.7 联合	(209)
8.8 枚举	(213)
8.9 类型定义 typedef	(215)
习题八	(217)
第九章 文件	(221)
9.1 文件概述	(221)
9.2 文件的打开与关闭	(223)
9.3 文件读写函数	(225)
9.4 文件检测函数	(232)
9.5 文件的存取	(234)
9.6 文件的更新	(238)

9.7 删除文件函数和文件更名函数	(238)
习题九	(239)
第十章 编译预处理	(245)
10.1 宏定义	(245)
10.2 文件包含	(249)
10.3 条件编译	(252)
习题十	(254)
第十一章 Turbo C 存储模式及大程序	(256)
11.1 80x86 处理器的编址方法	(256)
11.2 指针修饰: near、far 和 huge	(257)
11.3 存储模式	(259)
11.4 大程序	(261)
附录一 C 运算符的优先级与结合性	(266)
附录二 Turbo C 2.0 常用库函数及其标题文件	(267)
附录三 Turbo C 2.0 编译错误信息	(276)
参考文献	(280)

第一章 C 语言初步介绍

C 语言是一种通用的程序设计语言,它以表达简明、使用灵活、结构化的流程控制、丰富的数据结构和操作符集合、良好的程序可移植性和高效率的目标代码为特征。C 语言不仅具有高级语言的特点,还具有低级语言的功能,因此既可用于编写系统程序,也可用于编写不同领域的应用程序。现在,C 语言编译系统几乎可以在所有的操作系统以及各种类型的机器上运行,而且从个人计算机到大型计算机,有相当大比例的软件是用 C 语言编写的。许多原来用其他语言编写的较早的系统都改用 C 语言重新编写,这就是 C 语言广泛流行的原因。C 语言具有极强的兼容性和良好的用户界面,因而在程序设计语言的大奖赛中屡屡获胜,备受软件设计人员的青睐。

1.1 C 语言的起源和发展

C 语言的祖先是 ALGOL60(ALGOritm Language)。ALGOL60 是 1960 年由国际计算机委员会设计的一种面向过程的高级语言。它是一种结构化语言,用它编写的程序具有可读性和可移植性好的特点。但是,它不能直接对硬件进行操作,不宜用来编写系统程序。系统程序主要用汇编语言编写,而汇编语言是面向机器的,用它编写的程序可读性和可移植性都比较差。为此,人们开始考虑设计一种集高级语言和低级语言特点于一身的语言,以便能用它来编写可读性和可移植性都比较好的系统程序。

1963 年,英国的剑桥大学和伦敦大学首先将 ALGOL60 发展成 CPL(Combined Programming Language)语言。该语言已比较接近于硬件,但规模较大,难以实用。

1967 年,剑桥大学的 Martin Richards 将 CPL 改制成 BCPL(Basic Combined Programming Language)。BCPL 比 CPL 大为简化,既具有结构化程序设计语言的特点,也能直接处理与硬件相关的数据,被软件人员用作系统程序的描述语言。

1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 修改成 B 语言(Boiling CPL down to its basic good feature),并用 B 语言记叙和开发了第一个高级语言 UNIX 操作系统,在 DEC 公司的 PDP-7 小型机上实现。

1972 年,Ken Thompson 在 UNIX 系统上的亲密合作者 Dennis M. Ritchie 将 B 语言修改设计成 C 语言。C 语言既保持了 BCPL 和 B 语言的精练和接近于硬件的特点,也克服了它们过于简单、数据无类型等缺点。1973 年,Ken Thompson 和 Dennis M. Ritchie 又合作将 1969 年用汇编语言编写的 UNIX 操作系统改用 C 语言编写,其中 C 语言代码占 90% 以上,只保留了少量汇编语言代码。这样就使得 UNIX 操作系统向其他类型的机器上移植变得相当简单。到了 70 年代中期,UNIX 操作系统和 C 语言作为软件设计师的良好工具传遍了贝尔实验室,接着也传遍了几乎所有的美国大学校园。随着西欧和日本相继宣布加入 UNIX 和 C 的行列,UNIX 和 C 开始风靡世界。

C 语言的可靠性和易于应用使它有逐渐替代 UNIX 上其他程序设计语言的趋势。C 语言

是靠自身的优势发展起来的。它的用户越来越多,令人刮目相看。

1978年,以UNIX第7版中的C编译程序为基础,Brian W. Kernighan和Dennis M. Ritchie合著了影响深远的名著《The C programming Language》。这本书中介绍的C语言成为后来广泛使用的C语言版本的基础,称为K&R C语言。其后的十几年中,适用于不同机种和不同操作系统的C编译系统相继问世,从而把C语言的应用推向了更加广泛普及的阶段。然而,尽管C语言源程序兼容性很强,但由于没有统一的标准,各种版本之间略有差异。为此,美国国家标准局于1983年制定了C语言标准。这个标准不断完善,并从1987年开始实施,称为ANSI C。1988年,K&R修改了他们的经典著作《The C Programming Language》,按ANSI C标准重新编写了该书。现在一般称ANSI C为新标准,称K&R C为旧标准。此后陆续出现的各种C语言编译版本,如Microsoft C 5.0 9.0, Turbo C 2.0 3.0, Quick C等都是与ANSI C兼容的新版本。它们的语法和语句功能是一致的,其差异表现在各自的标准函数库中收纳的函数种类、格式和功能上,尤其是图形库的差异更大一些。

C语言是一种面向过程的语言,意思是用C语言编程时必须按照算法的实现过程逐条编写,通知计算机一步一步该怎么做。进入80年代后,面向对象的程序设计概念日益普及。C语言在发展的同时,也朝着成为支持面向对象的编程语言(OOP—Object - Oriented Programming Language)的方向迈出了步伐。1986年,美国AT&T公司的贝尔研究所推出了C语言的超集——C++语言。它已成为开发复杂大系统的具有吸引力的语言。由于C++语言是以C语言为基础的,所以,要学习C++语言,必须全面掌握C语言。

本书介绍的C语言将以ANSI C为基础,以ANSI在1988年10月31日提交的手稿,即“美国信息系统的国家标准——程序设计语言C, X3.159——1989”为依据。书中的程序是在Turbo C 2.0环境下调试的,使用其他C编译系统的读者请注意参考有关手册。

1.2 C语言的特点

在程序设计语言的大家族中,C语言技压群芳,具有强大的生命力,这主要是C语言本身的优势造成的,是C语言的特点决定的。

1. 语言表达能力强

C语言是通用的,不局限于某个机器或某个操作系统。它是一种“低级”的高级语言,也是一种“高级”的低级语言,它除了具有一般高级语言所能处理的算术和逻辑运算外,还可以直接进行通常由硬件实现的位、字节和地址等操作,因此在系统程序的设计中很有效,有“高级汇编语言”之称,以致足以取代汇编语言来编制各种系统软件和应用软件。

2. 语言简洁、使用灵活、易于学习和使用

C语言在表示方法上力求明了易懂、简单易行。例如,C语言的书写格式没有限制;用赋值运算符(如 $+=$ 、 $-=$ 、 $*=$ 、 $\div=$ 、 $++$ 、 $--$ 等)实现既运算又赋值;提供了预处理能力,包括宏定义、文件蕴含、条件编译等,可以在源程序编译时方便、灵活地进行修改;C语言还把一般语言的许多成分通过显式调用来完成,使得编译程序小而精。C语言并未提供依赖于硬件的输入/输出语句,也没有并行操作、同步等复杂控制,而是提供大量而有效的函数库来实现输入/输出、字符串处理、数学计算、动态存储分配等功能,函数库可根据需要方便地扩充。这些不仅保证了C语言移植性好,而且运行时所需支持少,存储空间占用也小。

3. C 语言基本上是一种结构化程序设计语言

C 语言具有顺序、选择、循环三种基本控制结构。它的各种控制语句如 if、while、for、switch、break 等,功能灵活,足以描述结构良好的程序。它还为被广为责难的 goto 语句提供了一席之地,兼顾了程序效率和简洁的要求。C 语言的代码和数据分隔特性(使用全局变量和局部变量)、代码块分隔特性(将一组语句放在花括号中)在功能上有助于数据隐蔽;C 函数的独立性有助于进行模块化程序设计。当然,C 语言不允许在函数中定义函数,所以它不是真正的块结构语言。

4. 具有很强的数据处理能力

C 语言可以在基本类型(整型、字符型、浮点型)的基础上按层次产生各种构造类型和集合类型,如数组、结构、联合、枚举等,利用它们可以处理各种复杂的数据结构。

5. C 语言生成的目标代码体积小、质量高

高级语言能否用来描述系统软件,特别是操作系统、编译程序等,除要求语言表达能力强之外,很重要的一个因素是语言生成的目标代码质量如何。如果目标代码质量低、系统开销大,就没有实用价值。许多实验表明,针对同一问题,用 C 语言编写的程序,其目标代码的效率仅比汇编语言低 10%~20%。由于用高级语言描述问题比汇编语言编程迅速、工作量小,可读性和可移植性好,易于调试和修改,所以 C 语言就成了人们描述系统软件和应用软件的比较理想的工具。

6. 可移植性好

C 语言是伴随 UNIX 操作系统诞生的。1978 年以后,C 语言已先后移植到大、中、小、微型计算机上,已独立于 UNIX 了。现在 C 语言编译系统几乎可以在所有的操作系统以及各种类型的机器上运行。统计资料表明,不同机器上的 C 编译程序有 80%的代码是公用的,因此移植起来很简便。

7. 绘图功能强

C 语言在庞大的图形程序库的支持下,可进行二维、三维图形和动画设计,也可用于计算机辅助设计。

C 语言具有上述诸多优点,因而具有旺盛的生命力,但也有一些缺点和不足。如运算符的优先级太多,不便于记忆,有些运算符的优先级还与常规约定不同;类型转换比较随便;检测手段太弱(如 C 语言不检查数组下标越界和变量类型的可兼容性等),安全性较差,要求程序员对程序设计更熟练、查错能力更强。但是,瑕不掩瑜,多种程序设计的应用已经证明 C 语言是一种极为有效且表达能力很强的语言,用 C 及 C++ 语言编写的软件所占比重愈来愈大的趋势仍将继续下去。

1.3 C 程序的基本结构

C 程序的基本结构是函数。一个 C 程序由一个或多个 C 函数组成。C 函数同 FORTRAN 语言中的程序段和 Pascal 语言中的过程相当,通常描述一个模块。一个 C 函数由若干条 C 语句组成。C 语句是完成某种程序功能的最小单位。一条 C 语句由若干个基本单词组成。

C 程序的一般格式如图 1-1 所示,其中 f1() 到 fn() 表示用户定义的函数。

1.3.1 基本单词

C语言共有五种基本单词,即关键字(亦称保留字)、标识符、常数、操作符和分隔符。例如语句

```
int x=5,y=x+3;
```

中,“int”是关键字,代表数据类型是整型;“x”和“y”是标识符(这里表示变量名);“=”“+”“,”是操作符,其中前两个是运算符,第三个是有特定用途的标点符号;“5”和“3”是常数;句首的“int”和“x”之间的空格是分隔符。

1. 关键字

关键字是C语言中有特定意义和用途、不得作为它用的字符序列,其中ANSI C标准规定的关键字有32个,Turbo C扩充的关键字有11个,见表1-1。

所有C关键字都用小写字母。在C语言中,大写字母和小写字母是有区别的。else是关键字,而ELSE就不是关键字。

2. 标识符

标识符是C语言中用来表示变量名、数组名、函数名、指针名、结构名、联合名、枚举常数名、用户定义的数据类型名及语句标号等的字符序列,可由1~32个字符组成。第一个字符必须是字母或下画线,后面的字符可以是字母、数字或下画线。例如AB、Ab、aB、ab、A_b、_ab、ab_s2d、W_length等都是正确的标识符,而A+B、A'B、A.B、2abc、α、β、d%等是错误的标识符。

表 1-1 Turbo C 关键字一览表

ANSI C 标准关键字			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while
Turbo C 扩充关键字			
asm	_cs	_ds	_es
_ss	cdecl	far	huge
interrupt	near	pascal	

标识符除了大小写不等价外,也不能与C关键字相同。例如ELSE可以作为标识符,它不

```

全局变量说明
main( )
    | 局部变量说明
    | 语句序列
    |
f1(形式参数表)
    | 局部变量说明
    | 语句序列
    |
f2(形式参数表)
    | 局部变量说明
    | 语句序列
    |
.
.
.
fn(形式参数表)
    | 局部变量说明
    | 语句序列
    |

```

图 1-1 C 程序的一般格式

会与 C 关键字 else 混淆, AB 和 Ab 是两个不同的标识符。

顺便指出,在 C 编译系统的库函数中,经常使用以下画线“_”打头的函数名或变量名,所以在程序中应尽量避免使用以“_”打头的标识符。

3. 常数

常数包括数值常数(如 123、-23.5、1.2E4 等)、字符常数(如 'a' 'B' 'c' 等)、字符串常数(如 "xyz" "good morning" 等)以及枚举常数(见第八章)。

4. 操作符

操作符包括各种运算符(如 +、-、*、/)和有特定意义的标点符号(如花括号、方括号、圆括号)等。

5. 分隔符

分隔符用来分隔相邻的标识符、关键字和常数,包括空格、制表符、换行符、换页符等。

1.3.2 C 语句

C 语句是组成 C 程序的基本单位,具有独立的程序功能。所有的 C 语句都以分号结尾。

1. 简单语句

任何 C 表达式加上分号后,就构成一条 C 语句。如

```
i = 0;
x = x + 1;
```

等。

2. 复合语句

一组 C 语句用花括号括住,就构成复合语句。如

```
do while(i < 10)
{
    sum = sum + i;
    i + +;
}
```

复合语句被视为一个整体,通常用在条件分支或循环语句中。有时为了数据隐藏的目的,复合语句可形成一个代码块,块中定义的局部变量不会对程序的其他部分产生副作用。

3. 空语句

只有一个分号的语句称为空语句,如

```
for(i = 0; i < 1000; i + +)
    ;
```

其中,空语句用作循环语句的循环体,表示什么也不做。事实上,这个循环的功能是延迟一小段时间。有时,空语句被用作转向点。

4. 注释

严格地讲,注释不是 C 语句。一个字符串以“/*”开始,而以“*/”结束称为注释,如

```
/* This is a main function */
```

注释既不被编译也不被执行,使用注释主要是为了增加程序的可读性。注释可以出现在程序的任何位置,允许分成若干行书写。

1.3.3 函数

C 函数是一个相对独立的程序段或模块。一个 C 程序由一个主函数和任意多个其他函数组成。

例 1-1 计算二整数相乘的程序。

```
/* ex-1.c */
#include <stdio.h>          /* 文件蕴含 */
main( )                    /* 主函数名 */
{                            /* 函数体开始 */
    int x,y,product;        /* 局部变量类型定义 */
    printf("enter the value of x,y:"); /* 屏幕提示信息 */
    scanf("%d%d",&x,&y);    /* 输入变量值 */
    product = x * y;        /* 计算乘积 */
    printf("x = %d, y = %d \n",x,y); /* 打印 x 和 y 的值 */
    printf("x * y = %d \n",product); /* 打印乘积值 */
}                            /* 函数体结束 */
```

这是最简单的 C 程序。根据程序中给出的注释,读者不难看出该程序的功能。其中,第一行是调用 C 编译程序提供的标题文件,因为下面用到的库函数 `scanf()` 和 `printf()` 都是在标题文件 `stdio.h` 中定义的。

程序运行时,屏幕首先提示信息:

enter the value of x,y: __

提示信息末尾的“__”是闪烁的光标。用户若从键盘输入 5 和 8(5 和 8 用空格隔开)并按回车键后,程序输出为

x = 5, y = 8

x * y = 40

C 程序的主要特点如下。

①一个 C 程序由一个或多个函数组成,其中有且仅有一个主函数,其他函数可以有多个,也可以没有。主函数有固定的名称 `main()`,其他子函数的名称由用户定义,取名规则与标识符相同。通常,主函数包括了整个程序的轮廓,由它再调用其他函数。主函数在程序中的书写位置可以任意,但不管主函数位于程序中的何处,程序运行时总是从主函数开始。

②在函数体内,主要有两大部分,第一部分是本函数内部用到的局部变量类型说明(C 语言中,所有的变量都要先说明后使用),第二部分是语句序列,完成本函数的功能。

③C 程序书写格式完全自由,既允许一行内写几条语句,也允许一条语句分几行书写,但每条语句都必须以分号(;)结束。为使程序清晰易读,通常每行写一条语句;不同结构层次的语句从不同的位置开始,即按缩进格式书写,可以用 Tab 键或空格键调整各行的起始位置;有时,还可以在程序的适当地方(如两个函数之间)加进一个或多个空行;使用注释可以明显增加程序的可读性。

1.3.4 库函数和标题文件

库函数不是 C 语言本身的组成部分,而是由 C 编译系统提供的一些非常有用的功能函

数。例如,C语言没有输入输出语句,也没有直接处理字符串的语句,但是C编译系统以库函数的方式提供了这些功能,另外还有大量的数学函数及其他函数。这些库函数的说明、类型和宏定义都保存在相应的标题文件(也称头文件)中,而对应的子程序则存放在运行库(.lib)中,用户可以直接调用。调用方式是在程序的函数外部用

#include <标题文件>

或

#include "标题文件"

包含指定的标题文件,就可以调用其中定义的库函数。例1.1程序中就是通过标题文件stdio.h调用了库函数scanf()和printf()来进行变量的输入和输出的。

Turbo C提供的标题文件如下(标题文件名不区分大小写):

ALLOC.H	动态分配函数(ANSI C)
ASSERT.H	诊断函数(ANSI C)
BIOS.H	ROM - BIOS 函数
CONIO.H	屏幕处理函数
CTYPE.H	字符处理函数(ANSI C)
DIR.H	目录处理函数
DOS.H	DOS 接口函数
ERRNO.H	错误码定义函数(ANSI C)
FCNTL.H	OPEN 函数使用的常数定义
FLOAT.H	与机器有关的浮点数定义及限制(ANSI C)
GRAPHICS.H	图形函数
IO.H	UNIX 用 I/O 子程序
LIMITS.H	与机器有关的各种限制(ANSI C)
LOCATE.H	国家和语言标识函数(ANSI C)
MATH.H	数学函数(ANSI C)
MEM.H	内存操作函数
PROCESS.H	spawn()和 exec()函数
SETJMP.H	非局部转移
SHARE.H	文件共享
SIGNAL.H	中断信号处理函数(ANSI C)
STDARG.H	可变长度参数表定义(ANSI C)
STDDEF.H	公用常数定义(ANSI C)
STDIO.H	标准输入输出函数(ANSI C)
STDLIB.H	标准实用函数(ANSI C)
STRING.H	字符串处理函数(ANSI C)
TIME.H	系统时间函数(ANSI C)
VALUES.H	与机器有关的常数
SYS\STAT.H	为打开文件定义的常数
SYS\TIMEB.H	为 ftime()函数的说明
SYS\TYPES.H	与时间函数联用的类型定义

其中,未注明(ANSI C)的都是由 Turbo C 定义的标题文件。

需要说明的是,不同的 C 编译系统提供的库函数在数量、种类、名称及使用上有一些差