

面向21世纪高等院校计算机规划教材

C语言 程序设计

邱富杭 曾健民 主 编

何庆新 王 方 李艳丽 周云玲 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

面向 21 世纪高等院校计算机规划教材

C 语言程序设计

邱富杭 曾健民 主 编

何庆新 王 方 李艳丽 周云玲 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书编者总结了多年计算机语言程序设计课的教学经验,打破以理论先行的传统方式,针对计算机程序设计初学者特点,采用任务驱动、范例领先的方式来编写,力求主干知识突出,语言表达流畅,具有良好的可读性和易学性。

本书内容主要包括C程序设计基础知识、控制结构与语句、数组、函数、编译预处理、指针、结构体和共用体、位运算、文件等。本书内容全面、层次清晰、编排合理,并对C语言的难点、重点和例子程序都做了详细的阐述。本书还有配套的学习指导书。

本书适合作为高等院校高级语言程序设计课程的教材。

图书在版编目(CIP)数据

C语言程序设计 / 邱富杭, 曾健民主编. —北京:
中国铁道出版社, 2011. 1
面向 21 世纪高等院校计算机规划教材
ISBN 978-7-113-12364-2

I. ①C… II. ①邱… ②曾… III. ①
C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第259669号

书 名: C语言程序设计
作 者: 邱富杭 曾健民 主编

策划编辑: 严晓舟 刘勇瑞

责任编辑: 侯 颖

封面设计: 付 巍

版式设计: 于 洋

读者热线电话: 400-668-0820

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)

印 刷: 北京新魏印刷厂

版 次: 2011年1月第1版 2011年1月第1次印刷

开 本: 787mm×1092mm 1/16 印张: 15.5 字数: 374千

书 号: ISBN 978-7-113-12364-2

定 价: 30.00元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社计算机图书批销部联系调换。

前 言

C 语言是目前国际上广泛流行的一种结构化的程序设计语言,它以强大的功能、丰富的数据类型、使用灵活、兼具面向硬件编程的低级语言特性及通用性强、可移植性好等语言特点成为当今软件开发的主流。许多原来用汇编语言书写的软件,都可用 C 语言来编写(如著名的 UNIX 操作系统就是用 C 语言编写的)。C 语言不仅适合于开发系统软件,而且也适用于开发应用软件和进行大规模科学计算。通过全面、深入、系统地学习用 C 语言编写程序,能够较全面地掌握程序设计方法,并养成良好的程序设计风格。

计算机教育面向应用,学生学习的主要目的是“应用”程序设计语言解决应用领域中的问题,不需细致地研究程序设计语言本身十分严格的语法和语义。基于这样的观点,并针对以上问题,编者通过认真分析和研究,并结合多年教学实践中的认识和体会,强化了对算法和程序设计基本思想的分析。此外,还针对大学一年级学生初始接触程序设计概念的特点,通过列举大量的应用实例,初步系统地介绍面向过程、面向模块和结构化程序设计的思想和方法。在此基础上,结合上机实践,使学生对程序设计有一个比较全面、系统的了解,为今后的学习打下扎实的理论基础。

本书以项目驱动、实例导入的方式深入浅出地讲解了 C 语言程序设计的基本方法。在叙述方法上尽量做到通俗易懂、深入浅出,充分考虑到初学者的知识层次和接受水平。另外,本书还通过大量的例题和习题帮助学生循序渐进地学习,使其尽快理解、消化和掌握各章节的内容。全书注重算法设计与程序设计的关联性,强化模块化程序的设计方法。

全书内容可分为三部分,共 9 章。第一部分为第 1 章,是入门知识,简单介绍 C 语言的基础知识,主要内容有 C 语言程序的基本结构、数据类型和数据的存储方式、基本的程序表达式。第二部分为第 2 章~第 4 章,是程序设计的基础部分,主要内容有描述程序算法的方法、程序语句的基本控制结构。掌握了第一、第二部分的内容,读者可以完成简单的程序设计。第三部分为第 5 章~第 9 章,是模块化程序设计的概念和实现的方法及应用实例,主要内容有函数、数组、指针、结构、文件、位运算等。通过对这三部分知识的学习,可以逐步认识模块化程序设计的思想,掌握模块化程序设计的方法。

本书适合作为高等院校高级语言程序设计课程的教材,也可供一般工程技术人员参考使用。

本书由闽南理工学院信息管理系组织编写,前言、第 1 章、附录由邱富杭编写,第

2章由曾健民编写，第3、4章由何庆新编写，第5、6章由王方编写，第7、8章由李艳丽编写，第9章由周云玲编写，承担编写任务的均是拥有多年教学经验的教师。全书由邱富杭、曾健民统稿。

本书在编写过程中得到了闽南理工学院迟岩院长、许沧海、许栋梁、王坚和许为勇等院领导的指导，得到张毅军、李燕威、王宽程、杨伟、杨英钟、林德贵、董庆伟、曹卿、郑新、洪锦霞和卓为玲等老师的大力支持和协助，在此一并表示衷心的感谢！中国铁道出版社对本书的出版给予了大力支持，相关编辑出色的工作给我们留下了深刻印象，在此也表示感谢。由于本书编者的水平有限，书中难免会有疏漏和不足之处，恳请广大读者不吝赐教。

编 者

2010年10月

目 录

第 1 章 C 语言基础..... 1	
1.1 C 语言的发展与特点..... 2	
1.1.1 C 语言的发展..... 2	
1.1.2 C 语言的特点..... 3	
1.2 简单的 C 语言程序..... 4	
1.2.1 几个简单的 C 程序..... 4	
1.2.2 C 程序的基本结构..... 5	
1.2.3 C 语言的基本符号与 词汇..... 6	
1.2.4 C 语言应用程序的编译与 运行..... 8	
1.3 C 语言集成开发环境..... 8	
1.3.1 Turbo C 集成开发环境 介绍..... 8	
1.3.2 Turbo C 集成开发环境的 使用..... 8	
1.3.3 其他 C 语言集成开发 环境简介..... 14	
1.4 基本数据类型..... 16	
1.5 标识符、变量和常量..... 18	
1.5.1 常量与变量..... 18	
1.5.2 整型数据..... 19	
1.5.3 实型数据..... 22	
1.5.4 字符型数据..... 24	
1.5.5 不同类型数据的 混合运算..... 26	
1.6 基本运算符、表达式及 运算的优先级..... 28	
1.6.1 运算符简介..... 28	
1.6.2 算术运算符和算术表达..... 29	
1.6.3 赋值运算符和赋值 表达式..... 30	
1.6.4 逗号运算符和逗号 表达式..... 32	
1.6.5 其他常用运算符..... 32	
1.6.6 运算符的优先级与 结合性..... 33	
1.6.7 常用数学函数与表达式..... 34	
1.7 程序范例..... 35	
本章小结..... 38	
习题..... 39	
第 2 章 简单 C 语言程序设计..... 43	
2.1 顺序结构程序设计..... 43	
2.2 标准输入/输出..... 44	
2.2.1 格式输出函数 printf()..... 44	
2.2.2 格式输入函数 scanf()..... 46	
2.3 字符数据的输入/输出..... 47	
2.4 字符串数据的输入/输出..... 47	
2.5 程序范例..... 48	
本章小结..... 50	
习题..... 50	
第 3 章 分支结构程序设计..... 54	
3.1 if 结构..... 54	
3.1.1 if 语句..... 54	
3.1.2 if...else 语句..... 56	
3.1.3 if 语句的嵌套..... 58	
3.2 switch 结构..... 62	
3.2.1 switch 语句..... 62	
3.2.2 break 语句在 switch 语句中 的作用..... 63	
3.3 程序范例..... 66	
本章小结..... 68	
习题..... 68	
第 4 章 循环结构程序设计..... 73	
4.1 for 语句..... 73	
4.2 while 语句..... 77	
4.3 do...while 语句..... 80	
4.4 用于循环中的 break 语句和 continue 语句..... 82	
4.4.1 break 语句..... 82	

4.4.2	continue 语句	83	本章小结	160
4.5	循环结构的嵌套	84	习题	160
4.6	goto 语句	89	第 7 章 指针	169
4.7	程序范例	91	7.1 指针的概念	169
本章小结	96		7.1.1 指针和指针变量	170
习题	97		7.1.2 指针变量的定义	170
第 5 章 函数与宏定义	104		7.1.3 指针变量的操作	170
5.1 函数的概念	104		7.2 指针运算	172
5.1.1 函数的定义	105		7.2.1 指针的赋值运算	172
5.1.2 函数的声明和调用	107		7.2.2 指针的加减运算	172
5.1.3 函数的传值方式	108		7.3 指针与数组	173
5.1.4 变量的作用域和存储 类型	111		7.3.1 指向数组的指针	173
5.2 内部函数和外部函数	116		7.3.2 通过指针引用数组元素	174
5.3 递归函数的设计和调用	118		7.4 指针与字符串	177
5.4 预处理	121		7.4.1 指向字符串的指针	177
5.4.1 宏定义	121		7.4.2 字符串指针变量与 字符数组的区别	177
5.4.2 文件包含	126		7.5 指针与函数	178
5.4.3 条件编译及其他	127		7.5.1 函数指针变量	178
5.5 程序范例	129		7.5.2 指针型函数	179
本章小结	130		7.6 指向指针的指针	181
习题	131		7.7 程序范例	182
第 6 章 数组	140		本章小结	183
6.1 一维数组的定义和初始化	141		习题	184
6.1.1 一维数组的定义	141		第 8 章 结构体和共用体	189
6.1.2 一维数组的初始化	142		8.1 结构体类型概述	190
6.2 一维数组的使用	142		8.1.1 结构体类型的特点	190
6.3 多维数组	146		8.1.2 结构体类型的定义	190
6.3.1 二维数组	146		8.2 结构体类型变量的定义和引用	190
6.3.2 多维数组的定义	147		8.2.1 结构体类型变量的 定义	190
6.4 字符数组	148		8.2.2 结构体类型变量的 初始化和引用	191
6.4.1 字符数组的初始化	149		8.3 结构体数组	192
6.4.2 字符串的输入	150		8.3.1 结构体数组的定义	192
6.4.3 字符串的输出	151		8.3.2 结构体数组的初始化	192
6.5 数组作为函数的参数	153		8.4 指向结构体类型数据的指针	193
6.5.1 数组元素作为函数的 参数	153		8.4.1 结构体指针变量的定义和 初始化	193
6.5.2 数组名作为函数的参数	154			
6.6 程序范例	156			

8.4.2 结构体指针的应用	194	9.1.1 文件的概念	208
8.5 结构体与函数	194	9.1.2 文件的操作	209
8.5.1 结构体变量作函数 参数	194	9.2 文件操作程序范例	219
8.5.2 结构体类型的函数	195	9.3 位运算	223
8.6 链表	196	9.3.1 按位取反运算	223
8.6.1 链表的概念	196	9.3.2 按位左移运算	223
8.6.2 链表的实现	196	9.3.3 按位右移运算	223
8.6.3 动态链表	196	9.3.4 按位与运算	224
8.6.4 链表的操作	197	9.3.5 按位或运算	225
8.7 共用体	198	9.3.6 按位异或运算	225
8.7.1 共用体的概念	198	9.3.7 复合位运算符	225
8.7.2 共用体变量的定义和 引用	198	9.4 位运算程序范例	226
8.8 程序范例	199	本章小结	227
本章小结	202	习题	227
习题	203	附录 A 7 位 ASC II 表	233
第 9 章 文件操作与位运算	208	附录 B 常用函数	234
9.1 文件操作	208	附录 C 常用 C 语言语句的语法格式	236
		附录 D C 语言颜色表	239
		参考文献	240

第 1 章

C 语言基础

C 语言是一门优秀的程序设计语言，其本身是高级语言，却又同时具有低级语言的某些优点，既可以用于编写系统程序，也可以用于编写应用程序。C 语言是函数式语言，编制 C 程序就是编写 C 函数。早期的 C 语言主要是用于开发 UNIX 系统，由于 C 语言的强大功能和各方面的优点逐渐为人们所认识，到了 20 世纪 80 年代，C 语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到广泛的应用，成为当代最流行的程序设计语言之一。随着计算机的普及和发展，C 语言在各个领域的应用越来越广泛，几乎各类计算机都支持 C 语言的开发环境，这为 C 语言的普及和应用奠定了基础。

C 语言程序中使用各种变量都应预先加以定义，即先定义后使用。对变量的定义可以包括三个方面：数据类型、存储类型、作用域。C 语言不仅提供了多种数据类型，还提供了构造更加复杂的用户自定义数据结构的机制。

本章介绍 C 语言的特点、程序组成、书写规则、上机运行和调试应用程序的方法以及基本数据类型、运算符和表达式。

本章内容

- C 语言的发展及特点。
- C 程序的基本结构。
- C 语言的基本符号与词汇。
- C 语言执行过程及集成开发环境。
- 基本数据类型及常量与变量的概念，整型数据、实型数据、字符型数据的使用。
- 运算符与表达式及运算的优先级。

学习目标

- 掌握 C 程序的基本结构、掌握 C 语言的基本符号与词汇。
- 掌握 Turbo C 集成开发环境的基本使用方法。
- 能够在 Turbo C 中编辑和运行最简单的 C 程序。
- 了解基本数据类型及其常量的表示法。
- 掌握变量的定义及初始化方法。
- 掌握运算符和表达式的概念。
- 理解自动类型转换和强制类型转换。
- 了解常用数学函数与表达式，能够将一般的数学算式转化为 C 语言表达式。

本章任务

(1) 编写并在 Turbo C 集成开发环境中运行一个简单的 C 程序, 程序的功能是求半径为 2.0 的圆面积。任务可以分解为两部分:

- 编写程序——了解 C 程序的基本结构。
- 学会在 Turbo C 集成开发环境中编译、连接和运行 C 程序。

(2) 在实际编程中, 有时会对不同数据类型的数据进行运算, 以及计算一些数学算式的值。本章要完成的主要任务是求一元二次方程的根。任务可以分解为两部分:

- 变量的定义及赋值。
- 将数学算式转换成 C 语言表达式。

1.1 C 语言的发展与特点

1.1.1 C 语言的发展

用于编写计算机程序的语言称为程序设计语言, 它是人与计算机之间进行信息交流的工具。从计算机诞生到今天, 伴随着计算机技术的飞速发展, 程序设计语言也在不断升级换代, 主要经历了面向机器(机器语言和汇编语言)、面向过程(高级语言)和面向对象(高级语言)几个阶段。

机器语言: 计算机发展初期使用的语言, 它由二进制的 0、1 组成, 计算机虽可直接执行, 但它面向机器, 可移植性极差, 现在已经很少用到。

汇编语言: 使用助记符(英文单词或单词缩写)表示指令代码(如用 ADD 表示加法运算), 以便于记忆。在执行时, 汇编语言源程序由汇编程序先将其转换为目标程序, 最后由连接程序把目标程序转换为可执行程序。

汇编语言的显著特点是用它编写的程序能直接对计算机底层硬件进行操作, 但由于它仍然面向机器, 用它编写程序难度仍然很大(须懂得计算机原理), 且维护十分困难, 可移植性也差, 故不适合初学者。

高级语言: 采用近似于数学的语言描述问题(如 QBASIC 语言), 面向过程, 与计算机机型无关, 所以高级语言容易记忆, 容易编程, 有很强的通用性。用高级语言编写的程序不能直接在计算机上运行, 必须先将它翻译成机器语言才能被计算机执行, 故高级语言执行速度较慢。

在 C 语言产生之前, 人们编写系统软件主要是使用汇编语言。由于用汇编语言编写的程序依赖于计算机硬件, 其可读性和可移植性都比较差; 而一般高级语言又不具备低级语言能够直观地对硬件实现控制和操作、程序执行速度快的特点。在这种情况下, 人们迫切需要一种既有一般高级语言特性, 又有低级语言特性的语言。于是, C 语言应运而生。

C 语言介于高级语言和低级语言(汇编语言)之间, 兼有二者之特点, 特别适合用于底层开发。

C 语言是一种编译性程序设计语言, 它与 UNIX 操作系统紧密地联系在一起。UNIX 系统是通用的、交互式的计算机操作系统, 它诞生于 1969 年, 是由美国贝尔实验室的 K.Thompson 和 D.M.Ritchie 用汇编语言开发成功的。

C语言的产生和发展与UNIX有很大的关系，C语言的前身是BCPL语言。1967年，英国剑桥大学的Martin Richard推出BCPL语言（Basic Combined Programming Language）。1970年，贝尔实验室的K.Thompson以BCPL语言为基础，开发了B语言，并用B语言编写了UNIX操作系统，在PDP-7计算机上实现。1972年，贝尔实验室的D.M.Ritchie在B语言的基础上设计出C语言，C语言既保持了BCPL语言和B语言的精练、接近硬件的优点，又克服了它们过于简单的缺点。

C语言的发展历程简述如下：

1972—1973年，美国贝尔实验室的D.M.Ritchie在B语言的基础上设计出C语言，当时的C语言只是为描述和实现UNIX操作系统的一种工作语言，且只在贝尔实验室内部使用。

1973年，K.Thompson和D.M.Ritchie两人合作，将UNIX 90%以上代码用C语言改写，即UNIX第5版。

1975年，UNIX第6版公布后，C语言突出的优点引起人们普遍注意。

1977年，出现了可移植的C语言。

1978年，UNIX第7版公布，K.Thompson和D.M.Ritchie以该版C编译程序为基础，合著*The C Programming Language*一书，该书所介绍的C语言，成为后来广泛使用C语言版本的基础，被称为标准C语言。

1983年，美国国家标准化协会（ANSI）根据C语言问世以来的各种版本，对C语言进行发展和扩充，并制定了新的标准，称为ANSI C。

1990年，国际标准化组织（ISO）制定了ISO C标准。

1972年以来，C语言几经修改和发展，出现了多个版本。目前在微机上广泛使用的版本有多个，各有特点，但它们一般都是以ANSI C为基础的，其中比较常用的版本有Microsoft C、Turbo C、Quick C、Borland C等。本书主要以ANSI C为基础介绍C语言，以Turbo C为调试环境，也可以在Visual C++ 6.0集成开发环境下处理C语言源程序。

1.1.2 C语言的特点

C语言作为世界上应用最广泛的计算机语言之一，有其不可替代的、吸引人的特点。C语言具有以下几个基本特点：

- C语言是结构化程序设计语言。C语言程序的逻辑结构可以用顺序、选择和循环三种基本结构组成，便于采用自顶向下、逐步细化的结构化程序设计技术。用C语言编制的程序具有简洁、紧凑、使用灵活方便、容易理解、便于维护等优点。
- C语言是模块化程序设计语言。C语言的函数结构、程序模块间的相互调用及数据传递和数据共享技术，为大型软件设计的模块化分解技术及软件工程技术的应用提供了强有力的支持。
- C语言具有丰富的运算能力。C语言除具有一般高级语言所拥有的四则运算及逻辑运算功能外，还具有二进制的位（bit）运算、单项运算和复合运算等功能。
- C语言具有丰富的数据类型和较强的数据处理能力。C语言不但具有整型、实型、双精度型，还具有结构、联合等构造类型，并为用户提供了自定义数据类型。此外，C语言还具有预处理能力，能够对字符串或特定参数进行宏定义。
- C语言具有较强的移植性。C语言程序本身并不依赖于计算机的硬件系统，只要在不同种类的计算机上配置C语言编译系统，即可达到程序移植的目的。

- C 语言不但具有高级语言的特性，还具有汇编语言的特点。C 语言既有高级语言面向用户、容易记忆、便于阅读和书写的优点；又有面向硬件和系统，可以直接访问硬件的功能。可以直接访问物理地址，能进行位操作，实现汇编语言的大部分功能。
- C 语言具有较好的通用性。C 语言有时也被称为中级语言，这并非贬义，因为它把高级语言的最佳成份同汇编语言的控制和灵活性巧妙地结合到了一起。C 语言既可用于编写操作系统、编译程序等系统软件，也可用于编写各种小型应用软件。
- C 语言也有其不足之处：主要是 C 语言语法灵活，在某种程度上降低了程序的安全性，对程序员要求较高；C 语言比较适用于底层开发和小型精巧程序的开发（如硬件驱动、手机应用软件等），不适合用于开发大型应用软件。

1.2 简单的 C 语言程序

1.2.1 几个简单的 C 程序

C 语言的源程序由一个或多个函数组成，每个函数完成一种指定的操作，所以又称函数式语言。下面观察以下三例，可以发现 C 语言程序的构成和书写规则。

【例 1.1】在屏幕上显示信息“Welcome!”。程序代码如下：

```
main()
{
    printf("Welcome!\n");
}
```

运行结果：

Welcome!

程序说明：

- C 程序由一系列函数组成，这些函数中必须有且只能有一个名为 main 的函数，这个函数称为主函数，整个程序从主函数开始执行。在例 1.1 中，只有一个主函数而无其他函数。
- 程序第 1 行中的 main 是主函数的函数名，main 后面的一对大括号是函数定义的标志，不能省略。
- 程序第 3 行的 printf() 是 C 语言的格式输出函数，在本程序中，printf() 函数的作用是输出括号内双引号之间的字符串，其中“\n”代表换行符。第 3 行末尾的分号则是 C 语句结束的标志。
- 程序第 2 行和第 4 行是一对大括弧，在这里表示函数体的开始和结束。一个函数中要执行的语句都写在函数体中。

【例 1.2】求一个整数的平方。程序代码如下：

```
main()
{
    int num,square;        /* 定义两个整型变量 */
    num=10;                /* 把 10 赋值给变量 num */
    square=num*num;        /* 计算 num 的平方，并将计算结果赋值给变量 square */
    printf("square=%d",square); /* 输出变量 square 的值 */
}
```

运行结果：

square=100

程序说明:

- 这个程序由一个主函数组成, 其中, 第3行的 `int` 表示定义变量类型为整型, 该行定义了 `num`、`square` 这两个整型变量。
- 程序第4行和第5行中的语句均为赋值语句, “=” 为赋值运算符, 其作用是将其右边的常量或表达式的值赋值给左边的变量。
- 第6行中的 “%d” 是输入/输出函数中的格式字符串, 在此表示以十进制整数的形式输出变量 `square` 的值。程序的运行结果中, “%d” 的位置被 `square` 变量的值 (即 100) 取代。
- 程序中多次出现的 “/*” 和 “*/” 是一对注释符, 注释的内容写在这对注释符之间。注释内容对程序的编译和运行不起任何作用, 其目的是为了程序的提高可读性。在必要的地方给程序加上注释是一个好习惯, 这使得程序易于理解, 而对程序的理解是进一步修改和调试程序的基础。

【例 1.3】 输入两个整数, 输出其中的最大值。程序代码如下:

```
main()
{
    int a,b,m;                /* 定义3个整型变量 */
    scanf("%d%d", &a, &b);    /* 输入两个整数到变量a和b中 */
    m=max(a,b); /* 调用max()函数求a和b的最大值, 并把函数的返回值赋值给变量m */
                                /* a和b为实际参数 */
    printf("max=%d",m);       /* 输出变量m的值 */
}
max(n1,n2)                    /* 定义max()函数, n1、n2为形式参数 */
int n1,n2; /* 定义形式参数的类型, 也可合并到函数头中: max(int n1,int n2) */
{
    int t;
    if(n1>n2)
        t=n1; /* 比较n1和n2的大小, 将最大值赋值给变量t */
    else
        t=n2;
    return t; /* 返回变量t的值 */
}
```

运行结果:

```
23 96 ✓
max=96
```

程序说明:

- 该程序由两个函数组成, 一个是 `main()` 函数, 一个是 `max()` 函数。`max()` 函数的功能是求两个整数的最大值, 而数据的输入和输出则在 `main()` 函数中实现。
- `main()` 函数和 `max()` 函数的定义是相互独立的。
- 程序第5行调用 `max()` 函数时, 分别把实际参数 `a` 和 `b` 的值传递给形式参数 `n1` 和 `n2`, 因此, 调用 `max()` 函数的结果是求得了 `a` 和 `b` 的最大值。

1.2.2 C 程序的基本结构

通过上面这三个例子, 可以对 C 程序的基本结构归纳如下:

(1) C 语言程序由函数构成

函数是构成 C 程序的基本单位, 即 C 程序由一个或多个函数组成, 其中必须有且只能有一个

名为 `main()` 的主函数。例如，在前面的例 1.1 和例 1.2 中，均只有一个 `main()` 函数，而在例 1.3 中，则有 `main()` 和 `max()` 两个函数。

(2) 函数的基本结构

```
void main()                /* 主函数说明 */
{
    变量定义;              /* 主函数体 */
    执行语句组;
}

函数类型 函数名 1(参数)    /* 子函数说明 */
{
    变量定义;              /* 子函数体 */
    执行语句组;
}

函数类型 函数名 2(参数)    /* 子函数说明 */
{
    变量定义;              /* 子函数体 */
    执行语句组;
}

...

函数类型 函数名 N(参数)    /* 子函数说明 */
{
    变量定义;              /* 子函数体 */
    执行语句组;
}
```

其中，函数名 1~函数名 N 是用户自定义的函数。

由此可见，一个完整的 C 程序应符合以下几点：

① C 程序以函数为基本单位，整个程序由函数组成。其中主函数 `main()` 是一个特殊的函数，一个完整的 C 程序必须有且只能有一个主函数，它是程序启动时的唯一入口，其特殊性表现在 C 语言程序执行时开始于主函数也结束于主函数，归根到底，其他函数均受调于主函数。也就是说，C 程序没有主函数，便不能执行。除主函数外，C 程序还可包含若干其他 C 标准库函数和用户自定义的函数。这种函数结构的特点使 C 语言便于实现模块化的程序结构。

② 用户自定义的函数由函数说明和函数体两部分组成。函数说明部分包括对函数名、函数类型、形式参数等的定义和说明；函数体包括对变量的定义和执行程序两部分，由一系列语句和注释组成。整个函数体由一对花括号括起来。

③ 语句是由定义符、运算符和数据按照 C 语言的语法规则组成的，每个语句完成一个特定的功能，语句以分号结束。

④ 有的函数在定义时，函数名后的小括号内有形式参数，如例 1.3 中的 `max()` 函数。{} 内则是由若干语句组成的函数体，每个语句必须以分号结束。C 语言的书写格式较自由，一行内可以写多个语句，一个语句很长时也可以分写在多行上。

1.2.3 C 语言的基本符号与词汇

任何程序设计语言都规定了一套自己的基本符号和词汇，C 语言也不例外。

1. C语言的基本符号集

C语言的基本符号集采用ASCII码字符集，包括：

- 大小写英文字母各26个。
- 10个阿拉伯数字0~9。
- 其他特殊符号，包括以下运算符和操作符：

+	-	*	/	%	<
<=	>	>=	==	!=	&&
	!	&		~	=
++	--	?:	<<	>>	()
[]	.	->	^	#	sizeof
+=	-=	*=	/=	%=	&=
^=	=	,			

2. C语言的词汇

(1) 标识符

程序中用来标识变量名、函数名、数组名、数据类型名等的有效字符序列称为标识符。简单地说，标识符就是一个名字。

标识符的构成规则如下：

① 标识符只能由英文字母（A~Z、a~z）、数字（0~9）和下画线（_）三类符号组成，但第一字符必须是字母或下画线。

例如，下面的标识符是合法的：

sum、Sum、n2、_average、a_3、student_2_name

下面是不合法的标识符：

num-1、a#3、2student、!sum_2、number.3

② 大写字母与小写字母含义不同（即C语言是对大小写敏感的），如sum、Sum、SUM表示三个完成不同的标识符。

③ 一般的C编译系统只取标识符的前8个字符为有效字符，而Turbo C则取标识符的前32个字符为有效字符。

④ 通常，命名标识符时应该做到“常用取简，专用取繁”。一般情况下，标识符的长度在8个字符以内就可以了。

(2) 关键字

关键字又称保留字，是C语言编译系统所固有的、具有专门意义的标识符。C语言的关键字有32个，一般用做C语言的数据类型名或语句名，如表1-1所示。

表 1-1 C语言关键字

描述类型定义	typedef、void
描述存储类型	auto、extern、static、register
描述数据类型	char、double、float、int、long、short、struct、union、unsigned、const、enum、signed、volatile
描述语句	break、continue、switch、case、default、if、else、do、for、while、goto、sizeof、return

说明：

① 所有关键字的字母均采用小写。

② 关键字不能再作为用户的常量、变量、函数和类型等的名字。

在学习了上述相关知识之后，通过例 1.4 来完成本章开篇提出的任务之一。

【例 1.4】编写程序：求半径为 2.0 的圆面积。程序代码如下：

```
#include <stdio.h>
main()
{
    float s,r;           /* 说明语句，定义实型变量 s、r */
    r=2.0;               /* 赋值语句 */
    s=3.14*r*r;          /* 计算面积 s */
    printf("s=%f\n",s) ; /* 函数语句 */
}
```

运行结果：

S=12.56000

1.2.4 C 语言应用程序的编译与运行

写好一个 C 语言源程序后，一般要经过编辑、编译、连接和运行才能得到程序结果。

① 编辑：在文本编辑器中，用 C 语言语法编写源程序代码，即 C 语言源程序 (*.c)。

② 编译：通过编译器将源程序转换为机器代码，生成目标程序，在 C 语言源程序的编译过程中，可以检查出程序中的语法错误。

③ 连接：C 语言是模块化设计语言，一个 C 语言应用程序可能由多个程序设计者分工合作编写，需要将所用到的库函数以及其他目标程序连接为一个整体，生成可执行文件 (*.exe)。

④ 运行：运行可执行文件后，可获得程序运行结果。

1.3 C 语言集成开发环境

1.3.1 Turbo C 集成开发环境介绍

C 语言有许多种编译器，这些编译器之间只有很小的差别，只要学会其中的一种，对其他几种就能很快适应。本节主要介绍在微机上较为流行的 C 语言版本——Turbo C。

Turbo C 是美国 Borland 公司的产品。Turbo C 具有图形库和函数库，具有查错及协处理器的仿真功能。Turbo C 2.0 是在 MS-DOS 操作系统上使用的编译系统，也可以在 Windows 操作系统下运行。

1.3.2 Turbo C 集成开发环境的使用

Turbo C 为 C 程序提供了编辑、编译、连接、运行的集成工作环境，具有运行速度快、编译效率高、易学易用、使用方便等特点。

(1) 启动 Turbo C，进入工作窗口

开机后，按如下步骤进入 C 的工作环境：

在 DOS 环境下启动 Turbo C，输入“Turbo C 所在的盘符\Turbo C 所在路径\tc”，例如 C:\TurboC2\tc，按 Enter 键，则会出现图 1-1 所示的编辑窗口。

也可以用 cd 命令使 Turbo C 程序所在的目录成为当前目录，例如“cd c:\TurboC2”，然后在含有 Turbo C 程序子目录的命令行上输入“tc”，并按 Enter 键。

在 Windows 环境下启动 Turbo C, 可以在图 1-2 所示的资源管理器中双击文件名为 TC.EXE 的应用程序; 或者进入 Windows 的 MS-DOS 方式 (选择“开始”→“程序”→“附件”→“命令提示符”菜单命令, 进入 DOS 方式窗口), 通过 DOS 命令, 如 cd\c:\TurboC2, 进入 Turbo C 所在目录, 再输入“tc”, 即可出现图 1-1 所示的 Turbo C 编辑窗口。



图 1-1 Turbo C 编辑窗口



图 1-2 在 Windows 资源管理器中启动 Turbo C 操作界面

(2) Turbo C 操作界面

Turbo C 集成环境的操作界面主要分成五个部分: 主菜单、编辑窗口、状态栏、信息窗口和功能键提示行。

- 主菜单位于窗口屏幕的顶部, 包括 8 个菜单: File (文件)、Edit (编辑)、Run (运行)、Compile (编译)、Project (项目)、Options (选项)、Debug (调试)、Break/watch (中断/监视)。每一个主菜单还有其子菜单, 分别用来实现各种操作。
- 编辑窗口: 在主菜单的下面, 正上方有 Edit 字样作为标志, 主要用于 Turbo C 源程序的输入和编辑。
- 状态栏: 在编辑窗口的上部有一行英文, 这一行英文提供正在编辑操作的信息, 其中 Line 1 和 Col 1 表示当前光标位置在第 1 行第 1 列, 标识光标当前所在的位置。当光标移动时, Line 和 Col 后面的数字也随之改变。Insert 表示当前处于插入状态。右端显示当前正在编辑的文件名, 对于新文件自动命名为: NONAME.C, 对于调入一个已存在的文件, 该位置上将显示调入文件的文件名。

- 功能键提示行: 在信息窗口的下面, 显示一些功能键的作用。

常见的功能键的功能说明如下:

F1——Help: 按 F1 键系统将会根据光标当前位置为线索给出帮助信息。

F5——Zoom: 扩大当前活动窗口 (编辑窗口或信息窗口)。

F6——Switch: 编辑窗口和信息窗口之间的切换。

F7——Trace: 用于跟踪程序的运行情况。

F8——Step: 按一次 F8 键执行一个语句。

F9——Make: 进行编辑和连接, 生成目标文件, 但不运行程序。

F10——Menu: 激活第一个菜单 File, 进入主菜单选择状态。