

□ 高等学校教材

大学C语言 程序设计教程

沈智慧 陈 青 梁爱南 饶居华 蒋 鸿 编著



高等教育出版社
HIGHER EDUCATION PRESS

高等学校教材

大学 C 语言程序设计教程

沈智慧 陈 青 梁爱南 饶居华 蒋 鸿 编著

高等教育出版社

内容简介

本书是根据教育部非计算机专业计算机课程教学指导分委员会制定的《非计算机专业计算机基础课程教学基本要求》和《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求（试行）》中提出的要求编写的，反映了高等学校非计算机专业计算机基础课程教学改革的新成果。本书的特点是强调实用性，注重教材的理论与实际相结合，以培养学生程序设计的基本方法和基本技能为目标。

本书共分 11 章，主要内容有 C 语言概述、简单的 C 程序设计、基本数据类型及表达式、选择结构、循环结构、函数及预处理、数组、指针、复杂结构类型、文件和 C++ 简介等。本书内容由浅入深，强化知识点、算法、编程的方法与技巧，为读者能够很好地学习 C 语言程序设计打开了方便之门。本书同时配套出版了《大学 C 语言程序设计实验教程》，包括四部分内容：集成环境简介、设计实验、课外实验、课程设计指导。

本书适合作为高等学校本科各专业“计算机程序设计”类课程的教学用书，也可作为全国计算机水平等级考试的教学与辅导用书，还可供广大程序开发人员参考。

图书在版编目（CIP）数据

大学 C 语言程序设计教程/沈智慧等编著. —北京：高等教育出版社，2009. 2
ISBN 978 - 7 - 04 - 026411 - 1

I. 大… II. 沈… III. C 语言 - 程序设计 - 高等学校 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字（2009）第 010435 号

策划编辑	刘 艳	责任编辑	康兆华	封面设计	王凌波	责任绘图	吴文信
版式设计	余 杨	责任校对	杨雪莲	责任印制	毛斯璐		

出版发行	高等教育出版社	购书热线	010 - 58581118
社 址	北京市西城区德外大街 4 号	免费咨询	800 - 810 - 0598
邮政编码	100120	网 址	http://www.hep.edu.cn
总 机	010 - 58581000		http://www.hep.com.cn
经 销	蓝色畅想图书发行有限公司	网上订购	http://www.landracom.com
印 刷	北京宏伟双华印刷有限公司		http://www.landracom.cn
		畅想教育	http://www.widedu.com
开 本	787 × 1092 1/16	版 次	2009 年 2 月第 1 版
印 张	16	印 次	2009 年 2 月第 1 次印刷
字 数	380 000	定 价	20.80 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

版权所有 侵权必究
物料号 26411 - 00

郑 重 声 明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010)58581897/58581896/58581879

传 真：(010)82086060

E-mail：dd@hep.com.cn

通信地址：北京市西城区德外大街4号

高等教育出版社打击盗版办公室

邮 编：100120

购书请拨打电话：(010)58581118

前 言

C 语言是当今世界上应用广泛、最具影响力的程序设计语言之一。C 语言整体结构紧凑，编辑方便，功能丰富，表达能力强，使用灵活，应用范围广，既具有高级语言的特点，又能够实现汇编语言中的大部分功能，而且用 C 语言编写的程序具有极强的可移植性。现在很多高等院校都把“C 语言程序设计”作为一门重要的课程，各类计算机考试都包含了 C 语言程序设计的内容。

教材是体现教学内容和教学方法的载体，是深化教学改革、全面推进素质教育、培养高素质人才的重要保证。好的教材源于教育教学观念的更新、教学改革和教学实践。虽然目前程序设计类的教材有很多，但是在笔者多年的教学实践中，发现适合应用型大学程序设计课程教学要求的教材数量较少。实现程序设计的教学从“知识传授导向”向“能力培养导向”转移，摆脱以课程为中心的经验办学的思路，研究、探索和科学地规划、建立应用型人才培养体系所需要的程序设计教材体系，已成为我国高等院校计算机课程教学改革的首要任务。

探讨如何使教师和学生轻松、愉快地完成 C 语言程序设计课程的教学和学习，具有重要的学术理论价值和社会实践意义。针对 C 语言比较难学的现实情况，我们对全书内容做了合理组织和精心安排，用简洁精练的语言和典型的例题帮助学生理解复杂的概念，每章内容都按照循序渐进的方式进行组织，由浅入深，相互呼应，引导学生掌握 C 语言的编程方法，提高他们的应用能力。本书共分 11 章，主要内容有 C 语言概述、简单的 C 程序设计、基本数据类型及表达式、选择结构、循环结构、函数及预处理、数组、指针、复杂结构类型、文件和 C++ 简介等。本书内容由浅入深，强化知识点、算法、编程的方法与技巧，在每一章最后均附有小结、思考与练习，为读者能够很好地学习 C 语言程序设计打开了方便之门。

本书是由长期从事高等院校计算机基础教学的教师合作编写的，是参编者多年教学经验和智慧的结晶。本书是以培养学生程序设计的基本方法和基本技能为目标，以应用能力为侧重点的特色鲜明的教材。既顾及 C 语言本身内容的完整性和知识的系统性，又对 C 语言进行清晰、全面的讲解。本书采用 Turbo C++ 3.0 作为语法规则，所有例题均在 Visual C++ 语言环境下上机调试并通过。

为方便教学，作者将与本书配套的多媒体电子教案放在高等教育出版社网站上，以便广大读者下载使用，也可向作者垂询，作者联系方式：shenzhihui@tom.com。

本书由湖南工业大学沈智慧、陈青、梁爱南、饶居华、蒋鸿老师编写，全书的框架设计和统稿、定稿工作由沈智慧老师完成。

在本书的写作过程中，我们对书稿进行了反复的修改，几易其稿，并得到了不少专家和任课教师的大力支持，他们为本书的编写提出了许多宝贵的意见和建议，朱文球、黄贤明、张建伟、刘强、张阿敏、杨旌老师也做了大量的工作，在此表示衷心的感谢。

本书同时配套出版《大学 C 语言程序设计实验教程》，提供课内与课外实验指导。实验指

II 前言

导部分以主要知识点为主线设计实验题目，兼具趣味性和实用性，并以任务驱动的方式指导学生完成实验程序设计。该实验教程还包含了综合应用实例用于课程设计，相信学习者一定会有全新的体验和感受。

由于编者水平有限，书中难免存在错误和不妥之处，恳请读者批评指正。

编者

2008 年 11 月

目 录

第 1 章 C 语言概述	(1)	函数	(22)
1.1 C 语言的发展史和特点	(1)	2.5.2 getchar 字符输入 函数	(23)
1.1.1 C 语言的发展史	(1)	2.6 复合语句与空语句	(23)
1.1.2 C 语言的特点	(2)	2.6.1 复合语句	(23)
1.2 C 语言程序的基本构成	(2)	2.6.2 空语句	(24)
1.2.1 基本符号	(2)	2.7 应用举例	(24)
1.2.2 C 语言程序结构	(3)	2.8 本章小结	(27)
1.3 C 语言程序设计方法	(5)	思考与练习	(28)
1.3.1 算法的概念	(5)	第 3 章 基本数据类型及表达式	(30)
1.3.2 算法的描述方法	(6)	3.1 C 语言数据类型	(30)
1.4 C 语言程序的运行	(7)	3.1.1 常量与变量	(30)
1.4.1 C 语言程序的运行 步骤	(7)	3.1.2 整型数据	(33)
1.4.2 编辑、编译与运行 C 语言程序的方法	(8)	3.1.3 实型数据	(34)
1.5 本章小结	(11)	3.1.4 字符型数据	(35)
思考与练习	(12)	3.2 运算符与表达式	(37)
第 2 章 简单的 C 程序设计	(13)	3.2.1 算术运算与增 1 减 1 运算	(37)
2.1 程序设计的基本结构 和方法	(13)	3.2.2 关系运算与逻辑 运算	(39)
2.1.1 程序的三种 基本结构	(13)	3.2.3 赋值运算与复合 赋值运算	(40)
2.1.2 结构化程序设计 方法	(15)	3.2.4 逗号运算与条件 运算	(41)
2.2 C 语言语句概述	(15)	3.2.5 位运算与求字节数 运算	(42)
2.3 赋值语句	(16)	3.2.6 混合运算的优先级	(43)
2.4 格式输入与输出	(17)	3.3 运算中的数据类型转换	(44)
2.4.1 printf 格式输出 函数	(17)	3.3.1 自动转换	(44)
2.4.2 scanf 格式输入 函数	(21)	3.3.2 赋值转换	(44)
2.5 字符输入与输出	(22)	3.3.3 强制类型转换	(45)
2.5.1 putchar 字符输出		3.4 本章小结	(46)
		思考与练习	(46)

第 4 章 选择结构 (49)	6.6 本章小结 (111)
4.1 if 语句及其嵌套..... (49)	思考与练习 (111)
4.1.1 if 语句的三种形式..... (49)	第 7 章 数组 (114)
4.1.2 if 语句的嵌套..... (53)	7.1 一维数组 (114)
4.2 if 语句的应用举例..... (55)	7.1.1 一维数组的定义 (114)
4.3 switch 语句与 break 语句..... (58)	7.1.2 一维数组的引用 (115)
4.4 switch 语句的应用举例..... (60)	7.1.3 一维数组的存储 结构与初始化 (115)
4.5 本章小结 (62)	7.1.4 应用举例 (116)
思考与练习 (62)	7.2 二维数组与多维数组 (118)
第 5 章 循环结构 (66)	7.2.1 二维数组的定义 (118)
5.1 goto 语句..... (66)	7.2.2 二维数组元素的 引用 (118)
5.2 do-while 语句..... (67)	7.2.3 二维数组的存储 结构与初始化 (118)
5.3 while 语句..... (69)	7.2.4 多维数组 (120)
5.4 for 语句..... (71)	7.2.5 应用举例 (122)
5.4.1 for 语句的一般形式 ... (71)	7.3 字符数组 (125)
5.4.2 for 语句的变化形式 ... (73)	7.3.1 字符数组的定义 (125)
5.5 continue 语句及循环的 嵌套..... (76)	7.3.2 字符数组的初始化 ... (125)
5.5.1 continue 语句 (76)	7.3.3 字符串的输入与 输出 (126)
5.5.2 循环的嵌套 (77)	7.3.4 字符串处理函数 (129)
5.6 循环应用举例..... (79)	7.3.5 应用举例 (130)
5.7 本章小结 (82)	7.4 本章小结 (132)
思考与练习 (82)	思考与练习 (133)
第 6 章 函数及预处理 (84)	第 8 章 指针 (134)
6.1 函数的定义与使用 (84)	8.1 指针概述 (134)
6.1.1 函数的分类 (84)	8.1.1 指针的概念 (134)
6.1.2 函数的定义 (85)	8.1.2 指针变量的定义与 初始化 (135)
6.1.3 函数的调用 (87)	8.1.3 指针变量的赋值 (136)
6.1.4 函数参数的传递 (89)	8.1.4 指针变量的引用 (137)
6.2 函数的递归调用..... (91)	*8.1.5 指向指针的指针..... (138)
6.3 变量的作用域与存储方式 ... (96)	8.2 指针与数组 (139)
6.3.1 变量的作用域..... (96)	8.2.1 指针与一维数组 (139)
6.3.2 变量的存储方式 (99)	*8.2.2 指针与多维数组..... (145)
6.4 编译预处理..... (103)	8.2.3 指针与字符串 (146)
6.4.1 宏定义 (103)	
6.4.2 文件包含 (105)	
6.4.3 条件编译 (106)	
6.5 函数应用举例..... (107)	

*8.2.4 指针数组	(149)	10.3.3 格式化读写函数	(192)
8.3 指针与函数	(152)	10.3.4 文件随机读写 函数	(193)
8.3.1 指针作为函数的 参数	(152)	10.4 文件的定位	(195)
8.3.2 指向函数的指针	(154)	10.5 文件的错误检测	(197)
8.3.3 返回指针值的函数 ..	(155)	10.6 应用举例	(198)
*8.3.4 void main 函数的 参数	(157)	10.7 本章小结	(204)
8.4 应用举例	(158)	思考与练习	(204)
8.5 本章小结	(163)	*第 11 章 C++简介	(207)
思考与练习	(164)	11.1 C++的产生和特点	(207)
第 9 章 复杂结构类型	(165)	11.1.1 C++的产生	(207)
9.1 结构体	(165)	11.1.2 C++的特点	(208)
9.1.1 结构体类型的定义、 说明和引用	(165)	11.2 C++对 C 的扩充	(208)
9.1.2 结构体数组	(167)	11.2.1 C++的输入输出	(208)
*9.1.3 结构体指针和链表 ..	(169)	11.2.2 变量的引用	(210)
9.2 共用体	(171)	11.2.3 函数重载	(213)
9.2.1 共用体类型及变量 的定义	(172)	11.2.4 带默认参数的 函数	(214)
9.2.2 共用体变量的引用 ..	(173)	11.2.5 new 和 delete 运算符	(215)
9.3 枚举类型	(174)	11.3 C++面向对象程序设计	(217)
9.4 用户自定义类型	(176)	11.3.1 类与对象	(218)
9.5 应用举例	(177)	11.3.2 构造函数与析构 函数	(220)
9.6 本章小结	(179)	11.3.3 继承与派生	(224)
思考与练习	(180)	11.3.4 多态性与虚函数	(226)
第 10 章 文件	(182)	11.4 本章小结	(231)
10.1 文件概述	(182)	思考与练习	(231)
10.1.1 文件的概念	(182)	附录	(232)
10.1.2 文件的分类	(182)	附录 1 常用字符与 ASCII 码 对照表	(232)
10.1.3 文件结构指针	(183)	附录 2 运算符的优先级和 结合方向一览表	(233)
10.2 文件的打开与关闭	(184)	附录 3 C 语言中的保留字及 其用途	(234)
10.2.1 文件的打开	(184)	附录 4 C 语言常用语法提要	(235)
10.2.2 文件的关闭	(186)	附录 5 C 库函数	(238)
10.3 文件的读写	(186)	参考文献	(245)
10.3.1 文件的字符读写 函数	(187)		
10.3.2 文件的字符串 读写函数	(189)		

第 1 章 C 语言概述

【学习目标】

1. 了解 C 语言的发展史和特点。
2. 了解 C 语言程序的基本构成要素。
3. 理解算法的概念，理解程序设计的思想和方法。
4. 了解 Visual C++ 6.0 集成开发环境的使用方法。

1.1 C 语言的发展史和特点

1.1.1 C 语言的发展史

自 1946 年世界上第一台电子计算机问世以来，程序设计语言从机器语言、汇编语言、面向过程的程序设计语言和面向对象的程序设计语言先后经历了 4 个发展阶段。在目前数百种程序设计语言中，C 语言仍然是国际上广为流行的程序设计语言。

C 语言的前身是 ALGOL 语言。1960 年，ALGOL 60 版本推出，受到程序设计人员的普遍欢迎。用 ALGOL 60 来描述算法很方便，但是它距离计算机硬件系统很远，不宜用来编写系统程序。1963 年，英国剑桥大学在 ALGOL 语言的基础上增添了硬件处理能力，并命名为 CPL (combined programming language, 复合型程序设计语言)。CPL 由于规模庞大，学习和掌握比较困难，未能流行开来。1967 年，剑桥大学的马丁·理查德 (Martin Richard) 对 CPL 语言进行了简化，推出 BCPL (basic combined programming language, 基本复合型程序设计语言)。1970 年，美国贝尔实验室的肯·汤普逊 (Ken Thompson) 对 BCPL 进行了进一步的简化，突出了硬件处理能力，并取“BCPL”的第一个字母“B”作为新语言的名称，同时用 B 语言编写了 UNIX 操作系统。1972 年，贝尔实验室的布莱恩·W. 卡尼汉 (Brian W. Kernighan) 和丹尼斯·M. 利奇 (Dennis M. Ritchie) 对 B 语言进行了完善和扩充，在保留 B 语言强大硬件处理能力的基础上，扩充了数据类型，恢复了通用性，并取了“BCPL”的第二个字母作为新语言的名称。此后，两人合作，重写了 UNIX 操作系统。C 语言伴随着 UNIX 操作系统成为一种颇受欢迎的程序设计语言。

1977 年，为了让 C 语言脱离 UNIX 操作系统，成为在任何计算机上都能运行的通用程序设计语言，卡尼汉和利奇撰写了《C 程序设计语言》一书，对 C 语言的语法进行了规范化描述，成为当时的业内标准。随着微型计算机的普及，出现了 C 语言的不同版本，为了统一标准，美国国家标准化学会于 1987 年制定了 C 语言的标准，称为“ANSI C”。通常将卡尼汉和利奇的标准称为旧标准，将“ANSI C”称为新标准。

目前在微型计算机上使用的 C 语言编译系统有：Turbo C、Turbo C++ 3.0、Microsoft Visual C++ 6.0、Quick C 等。现在一般以“ANSI C”为标准，以 Visual C++ 6.0 为编译系统。

1.1.2 C语言的特点

C语言的主要特点可以概括如下。

(1) 丰富的数据类型

C语言不仅有基本数据类型，而且还有构造类型、指针类型等，能够用来实现各种复杂的数据结构。

(2) 丰富的运算符

C语言的运算符相当多，许多操作都可以用运算符来表示。由运算符和运算对象可以组成表达式，因而C语言中表达式的类型也是相当丰富的。

(3) 结构化的控制语句

在C语言中，三种类型的结构化控制都有相应的语句加以体现。因此可以用C语言进行结构化程序设计。C语言还以函数作为程序的基本单位，便于实现程序的模块化。允许编程者采用缩进格式编程。因此，用C语言设计的程序层次结构清晰。

(4) 具有位操作的功能，可以直接对硬件进行操作

这使得C语言既具有高级程序设计语言的所有优点，又具有低级语言的许多功能，成为所谓“中间语言”。用C语言编写的程序质量好，目标代码的执行效率高。

(5) C语言简洁紧凑

C语言只有32个保留字和9种控制语句，而且书写形式自由，所以C语言的入门学习比较容易。

(6) C语言是一种模块化的程序设计语言

C语言程序由一个或多个函数构成。C语言的主要结构成分为函数，函数可以在程序中被定义用来完成独立的任务，被单独编译成代码，以实现程序的模块化。

(7) 目标代码的质量高，执行速度快

C语言既可以用来编写复杂的系统软件，又可以用来编写应用软件。正因为C语言拥有众多优点，才能风靡全球，成为世界上应用最广泛的高级程序设计语言之一。

1.2 C语言程序的基本构成

1.2.1 基本符号

C语言的合法字符集有：

(1) 字母

小写字母a~z共26个，大写字母A~Z共26个。

(2) 数字

0~9共10个数字。

(3) 空白符

空格符、制表符、换行符等统称空白符。空白符只在字符常量和字符串常量中才起作用。在其他地方出现时，空白符只起间隔作用，编译程序将忽略它们。因此在程序中使用空白符与否，对程序的

编译不会发生任何影响，但在程序中的适当位置使用空白符将提高程序的清晰性和可读性。

(4) 特殊字符

特殊字符包括+、-、/、=、(、)、{、}、[、]、*、#、&、|、\、%、^、<、>、?、,、.、'、"、_、!、~、; 等。

此外，其他字符都只能放在注释语句、字符型常量、字符串型常量和文件名中。

除了这些合法字符之外，C 语言还包含一些词汇，如标识符、保留字、运算符和分隔符。

① 标识符

在 C 语言中，标识符是对变量、函数、标号和其他各种用户自定义对象的命名。标识符可以包含一个或多个字符。标识符的第一个字符必须是字母或下划线，随后的字符可以是字母、数字或下划线。

② 保留字

保留字又称关键字，是 C 语言中具有特定含义的单词。对保留字不能重新定义，也不能用作一般的标识符。C 语言的 32 个保留字如下：

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

③ 运算符

运算符是用来表示运算的某种符号。大多数运算符由一个字符构成，也有的运算符由多个字符构成。这将在后续章节中予以介绍。

④ 分隔符

分隔符是用来分隔数据、变量、表达式的符号。C 语言中常用的分隔符是空格符、逗号、制表符和换行符等。

1.2.2 C 语言程序结构

学习某种程序设计语言，最有效的途径就是用它来编写应用程序。下面给出几个简单的程序，读者将对 C 语言有一个初步的了解。

先看一个简单的 C 程序示例，了解 C 程序的基本构成要素。

例 1.1：将华氏温度转换为摄氏温度，并显示出来。华氏温度由键盘输入。

```
#include <stdio.h>           /*预编译命令*/
void main( )                 /*主函数*/
{
    float F,c;               /*声明部分，定义 2 个浮点型变量 F 和 c*/
    scanf("%f",&F);          /*执行部分，键盘输入值存放到变量 F 所在的存储单元中*/
    c=5*(F-32)/9;            /*执行部分，计算 5×(F-32)÷9 的值，并将计算结果赋给变量 c*/
}
```

```
printf("c=%.2f",c);    /*执行部分，向输出设备（显示器）输出变量c的值*/  
}
```

程序运行时，从键盘输入：

32✓（✓表示按回车键）

运行结果为：

c=0.00

例 1.2：求两个数的和。

```
#include <stdio.h>                                /*预编译命令*/  
int add(int x, int y)                             /*自定义函数 add*/  
{  
    return(x+y);  
}  
  
void main( )                                       /*主函数 main*/  
{  
    int a,b,sum;                                  /*声明整型变量 a、b 和 sum*/  
    sum=0;  
    printf ("Enter value of a \n");              /*显示提示信息*/  
    scanf ("%d", &a);                             /*读取一个整数并将其存入变量 a 所在的内存单元*/  
  
    printf ("Enter value of b \n");              /*显示提示信息*/  
    scanf ("%d", &b);                             /*读取一个整数并将其存入变量 b 所在的内存单元*/  
  
    sum=add (a,b);                                /*调用 add 函数求和*/  
    printf ("sum is %d \n",sum);                  /*输出结果*/  
}
```

程序运行时，屏幕显示提示信息：

Enter value of a

从键盘输入：

20✓

屏幕显示提示信息：

Enter value of b

从键盘输入：

30✓

运行结果为：

sum is 50

通过上面的两个例子可以看出:

① 一个 C 语言程序可以由多个函数组成,但必须包含且只能包含一个主函数 `main`, `void main()` 称为函数头,其后由花括号对“{}”括起来的部分称为函数体。程序总是从 `main` 函数的“{”后面的第一条可执行语句开始执行,到“}”结束。函数是组成 C 语言程序的基本单位。一个函数由函数头和函数体两部分组成。

(a) 函数头:定义了函数名、函数返回值类型、函数形式参数名及其数据类型。

例如:

```
int add (int x, int y)
```

(b) 函数体:

函数头下面花括号“{}”内的部分是函数体。函数体定义了该函数所要实现的功能。函数体一般包括:

(i) 说明部分。如例 1.2 中 `void main()` 函数中的“`int a,b,sum;`”是变量说明。

(ii) 执行部分。由若干条语句组成,每条语句都以分号“;”结束。

② 一个 C 语言程序中函数的出现次序可以是任意的,但总是从 `void main()` 函数开始执行。

③ 前面带有“#”的行,如“`#include`”、“`#define`”等都是编译预处理命令,也称为命令行。它不是 C 语言语句,所以后面不加分号“;”。

④ C 语言程序的书写格式比较自由,一行内可有多条语句,一条语句也可以占多行(但不能将一个单词分开)。为了提高程序的可读性,往往以缩进写法来体现语句的层次。

⑤ C 语言以小写字母作为基本书写形式,并区分字母大小写。

⑥ “/*”与“*/”之间的内容构成 C 语言程序的注释(在 Visual C++ 环境下也可以用“//”后跟内容表示注释)。“/*”与“*/”之间的内容既可以是一行,也可以是多行,但“//”后跟的注释内容只能写在同一行内。注释部分不参与程序的编译和执行,只起到说明的作用,增加程序的可读性。读者在编程时应根据实际情况进行标注。

1.3 C 语言程序设计方法

在对 C 语言程序有了初步的了解之后,可以开始学习 C 语言程序设计的方法。当面对一个编程问题的时候,首先要对程序进行分析,找出解决问题的步骤,再根据这些步骤编写程序。这里提到的解决问题的步骤就叫做“算法”。学习程序设计语言,关键就是设计算法。一旦有了正确的算法,用任何语言进行编码都不会有困难。下面对算法详细加以说明。

1.3.1 算法的概念

著名计算机科学家尼克劳斯·沃思(Niklaus Wirth)提出了一个有名的公式:

算法+数据结构=程序

这个公式的重要性在于它说明了算法是程序的灵魂,数据结构是程序加工的对象。

算法是对特定问题求解步骤的一种描述,它是指令的有限序列,其中每一条指令表示一个或多个操作。此外,一个算法还应具有以下 5 个重要特性:

(1) 有穷性

一个算法必须保证执行有穷步之后结束。

(2) 确定性

算法的每一条指令必须具有确切的含义，读者理解算法时不会产生歧义。也就是说，对相同的输入只能得到相同的输出。

(3) 可行性

一个算法应该是能够执行的，如“计算 $X/0$ 的值”是不允许的。

(4) 输入

一个算法可以有 0 个或多个输入。

(5) 输出

算法具有一个或多个输出。

1.3.2 算法的描述方法

由此可知，算法是对问题求解步骤的描述，考虑的是进行某一个问题求解的方法和步骤，是解决问题的框架和流程。设计一个算法，或者描述一个算法，最终要由程序设计语言来实现。一般可以使用下面几种类型的工具来描述算法。

1. 自然语言

自然语言即人们日常进行交流的语言，如汉语、英语或其他语种等。自然语言描述的算法通俗易懂。但是用计算机直接处理自然语言比较困难，而且容易产生歧义，对于含有分支和循环结构的算法也不便描述。所以，一般不用自然语言来描述算法。

2. 专用工具

要对某一个算法进行描述，可以借助有关的图形工具或符号。常用的工具有流程图、N-S图、伪代码、程序设计语言等。

流程图是一种传统的算法表示方法，它利用几何图形的框来代表各种不同性质的操作，用流程线来指示算法的执行方向。由于流程图简单、直观，所以应用广泛。特别是在早期程序设计语言阶段，只有通过流程图才能简明地表述算法，流程图成为程序员间交流信息的重要手段。图 1.1 所示为流程图常用的符号。

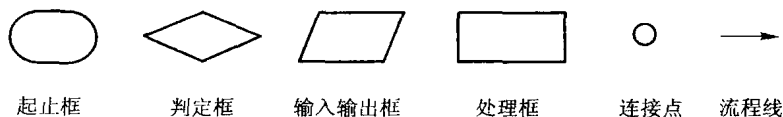


图 1.1 流程图常用符号

例 1.3: 用流程图表示求两个数中较大数的算法。

所设计的流程图如图 1.2 所示。

3. 伪代码表示

伪代码用介于自然语言和程序设计语言之间的文字和符号来描述算法。它如同一篇文章，自上而下地按顺序写下来。每一行（或多行）表示一个基本操作。伪代码书写方便，格式紧凑，也易于理解。

例如：求两个数中的较大数。用伪代码表示算法如下：

置 $a=1$;

置 $b=2$;

如果 $a>b$ ，则

$\max=a$;

否则

$\max=b$;

打印 $\max$ 的值。

4. 程序设计语言表示

用程序设计语言表示算法必须严格遵循所用语言的语法规则，实际上这就是一个完整的程序，这和伪代码是不同的。

例 1.4：求两个数中的较大数。用 C 语言表示的算法如下：

```
#include <stdio.h>

void main( )
{
    int a,b,max;
    scanf("%d,%d",&a,&b);
    if(a>b)
        max=a;
    else
        max=b;
    printf("a、b 两数中的较大数是%d\n", max);
}
```

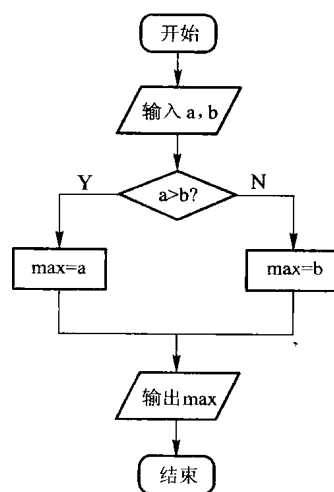


图 1.2 求两个数中较大数的流程图

1.4 C 语言程序的运行

1.4.1 C 语言程序的运行步骤

开发一个 C 语言程序，包括以下步骤。

(1) 算法设计

针对具体的问题，分析、建立解决问题的物理或数学模型，并将解题方法采用某种方式加以描述，为进行实际的 C 语言编程打下良好的基础。

(2) 程序编辑

程序员将编写好的 C 语言程序输入计算机，并将其保存为文件扩展名为“.c”的文件。编辑的结果是建立 C 源程序文件。C 语言程序习惯上使用小写字母，常量和用途的符号可使用大写字母。C 语言对大小写字母是敏感的。C 语言中的保留字必须用小写字母。

(3) 程序编译

将编辑好的源程序文件翻译成二进制目标代码。编译时，编译器首先要对源程序中的每一条语句检查是否存在语法错误。一旦发现错误，就在屏幕上显示错误所在的位置和错误类型等

信息。此时,要再次调用编辑器进行查错和修改。然后,再进行编译,直至排除所有语法和语义方面的错误。正确的源程序文件经过编译后会在磁盘上生成目标文件,其扩展名为“.obj”。

(4) 程序连接

把目标文件和系统提供的标准库函数连接在一起,生成可以运行的可执行文件。这一阶段仍有可能发现程序中的错误,并在屏幕上加以显示,必须对这些错误进行改正,再重新进行编译和连接。如果程序中没有错误,则系统生成可执行文件,其扩展名为“.exe”。

(5) 程序运行

在操作系统的控制下运行可执行文件。若得到正确的运行结果,则C语言程序的开发工作到此完成。否则,要进一步检查并修改源程序,重复编辑—编译—连接—运行的过程,直到取得正确结果为止。

为了编辑、编译、连接和运行C语言源程序,必须要有相应的C语言编译系统。目前常用的C语言编译系统有Turbo C 2.0、Turbo C++ 3.0和Visual C++ 6.0等。

微软公司于1997年和1998年推出的Visual C++ 5.0/6.0是集程序编辑、编译、连接、运行、调试于一体的C/C++语言集成开发环境。它使用方便,支持面向对象的C++语言成分,是Windows环境下的可视化开发工具,常用于进行C/C++语言程序开发的教学。

下面介绍在Visual C++ 6.0环境中输入、编辑、编译、连接和运行C语言程序的方法,目的是供读者学习C语言程序的编写与运行、调试,并不涉及C++的内容。所介绍的是一种简单的方法,回避了有关工程的概念。详细的使用方法请参看与本教材配套的《大学C语言程序设计实验教程》,可以学习和掌握更多的关于Visual C++集成开发环境的知识。

1.4.2 编辑、编译与运行C语言程序的方法

1. 启动 Visual C++集成开发环境

从“开始”菜单或通过快捷方式进入Visual C++集成开发环境后,将出现如图1.3所示的主界面。

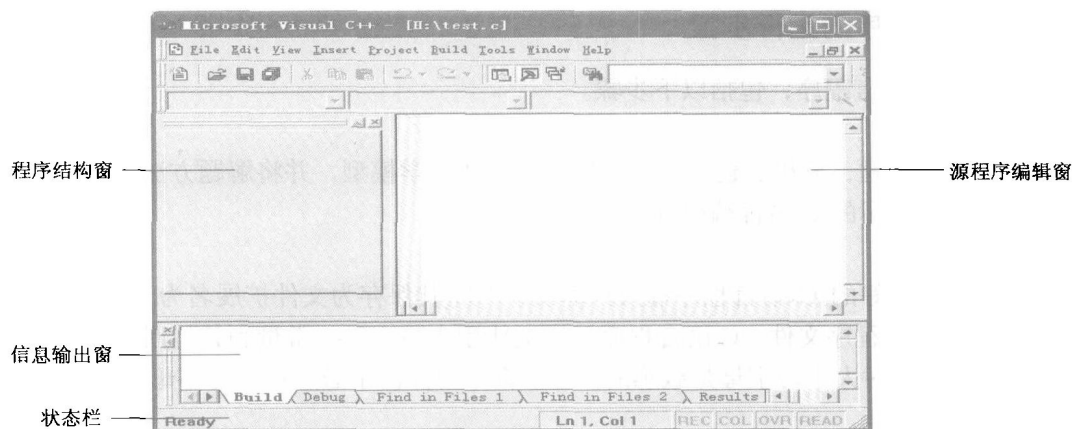


图 1.3 Visual C++主界面