

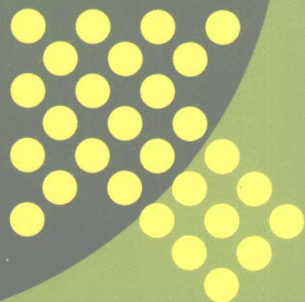
21世纪高等学校规划教材



C YUYAN CHENGXU SHEJI

C语言程序设计

梁 栋 李新华 徐 超 主编



中国电力出版社

<http://jc.cepp.com.cn>

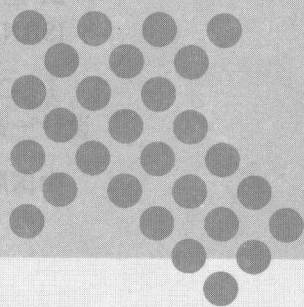
21世纪高等学校规划教材



C YUYAN CHENGXV SHEJI

C语言程序设计

主 编 梁 栋 李新华 徐 超
编 写 迟成文
主 审 李龙澍



中国电力出版社

<http://jc.cepp.com.cn>

内 容 提 要

本书为 21 世纪高等学校规划教材。全书共分 10 章, 主要内容包括 C 程序设计概述、数据及运算、流程控制、数组、指针、函数、结构体文件、C 语言实际应用、C 语言程序设计常见错误及解决方法等。针对 Visual C++ 6.0, 书中精心安排了三个简单实用的应用实例。

本书采用 Visual C++ 6.0 作为语言开发环境, 系统地介绍了 C 语言程序设计的知识。书中语言通俗易懂、简洁流畅; 内容叙述深入浅出、突出重点; 精选大量典型例题, 讲解由浅入深, 细致详尽; 独特的“想一想”进一步引导读者独立思考, 培养从程序设计的角度思考和解决问题的能力; 每一章的开头都有学习要点和学习难点介绍, 每一章结尾都有小结来提纲挈领, 强化知识点、编程方法和常用算法。习题类型丰富, 涵盖全部关键知识点。与本书配套的《C 语言程序设计习题解答与上机指导》不仅对全部习题给出了解答, 还安排了相应的上机实践和模拟试卷。

本书作为 C 语言程序设计省级精品课程主讲教材, 不仅可以作为普通高校本、专科学生学习 C 语言的教材, 也可以作为相关工程技术人员的自学与参考用书。

图书在版编目 (CIP) 数据

C 语言程序设计 / 梁栋, 李新华, 徐超主编. —北京: 中国电力出版社, 2009

21 世纪高等学校规划教材

ISBN 978-7-5083-8433-7

I. C… II. ①梁… ②李… ③徐… 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 012841 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://jc.cepp.com.cn>)

航远印刷有限公司印刷

各地新华书店经售

*

2009 年 2 月第一版 2009 年 2 月北京第一次印刷

787 毫米×1092 毫米 16 开本 23 印张 558 千字

定价 36.80 元

敬 告 读 者

本书封面贴有防伪标签, 加热后中心图案消失
本书如有印装质量问题, 我社发行部负责退换

版 权 专 有 翻 印 必 究

序

程序设计是目前大学本、专科学生需要掌握的一种技能，理工科学生多数选择 C 语言作为程序设计的语言。学生通过对 C 语言程序设计的学习，不但能掌握一种操控计算机的能力，而且可以使用这种技能来完成其他课程的计算性习题与实验中的数据处理，还能锻炼自己严密的逻辑思维能力、培养自己一丝不苟的科研精神。

C 语言程序设计的教材比较多，内容也很广泛，章节顺序的安排也有不同。本书的作者精心选取了最主要的必备知识点，采用比较顺畅的章节安排，使得知识点串连得比较合理。概念清楚准确，前后呼应，逐步深化和拓展，适合读者由简到繁、由浅入深的学习习惯。行文流畅、通俗易懂，便于读者学习掌握。配备的大量例题，既可以帮助读者加深对概念的理解，又具有实用程序的雏形，对于读者把 C 语言程序设计能力应用到其他课程很有指导性的作用。章末习题数量适中，从内容上看，既有例题的模仿，也有例题的拓展，还有少量考察读者算法设计和程序设计独创性能力的题目，对于读者掌握基本知识，开拓程序设计的能力均有帮助。

本书的作者中既有承担过全国自学考试统编教材的主编，也有写过多本 C 语言程序教材的作者，还有长期使用 C 语言开发软件的科研人员，所有作者均多年从事 C 语言程序设计课程的主讲工作。相信他们的丰富经验和辛勤劳动一定会减轻读者学习的难度，提高学习的效果，增加学习的兴趣。

希望本书的所有读者能不断地“挑刺”、并将所有的“刺”反馈给作者，也希望作者能不断修改和完善本书，使其最终成为一本真正的“精品教材”。

中国科学院院士

陆. 国. 良

2008 年 12 月

前 言

C 语言以功能丰富,表现力强,实用方便灵活,目标程序效率高,可移植性好一举成为国内外软件开发中的一种主流计算机语言。通过 C 语言不仅开发出了 Windows、UNIX 等操作系统和面向底层硬件的应用程序,还派生出了 C++、Java、C# 等高级程序设计语言。学好 C 语言,就可以很容易通过进一步学习掌握这些派生语言。学习并掌握 C 语言来理解计算机程序设计,培养计算机编程解决问题的能力,已为广大计算机从业人员和高等院校学生的迫切需求。

走进书店,C 语言的书籍琳琅满目。但随着 2007 年版全国计算机等级考试二级 C 语言程序设计考试大纲的推出,基于 Visual C++ 6.0 集成开发环境的 C 语言书却为数不多,一些 VC 版 C 语言书上的程序甚至会在 Visual C++ 6.0 环境下运行出错。有鉴于此,编者组织省级精品课程中长期从事 C 语言程序设计课程的主讲工作、多年使用 C 语言开发软件,具有丰富的教学经验和写作经验,出版过 C 语言全国统编教材的教师,精心编排了本书的内容。在书中的第 1、2 章,介绍了程序设计及算法的基本概念和 C 语言的基础知识;第 3、4、5 章介绍了 C 语言的基本程序设计技术,指向内存中成批数据的数组,借助内存地址间接访问数据的指针;第 6 章详细讨论了 C 语言程序的基本单位——函数,着重介绍了函数的各种调用、数据传递和变量的存储特性,避免了 C 语言学习中先讲函数容易导致概念模糊甚至出错的老毛病;第 7 章介绍了根据具体问题构造具体数据类型的结构和链表技术,第 8 章介绍了 C 语言文件的操作方法;第 9 章针对 Visual C++ 6.0,精心策划了三个简单实用的应用实例,最后就 C 语言程序设计的常见错误进行了分析。读者通过阅读本书,不仅可以轻松掌握 C 语言基本编程技术,还可以在本书一步步详细引导下编写基于对话框的 Windows 应用程序。

本书本着打造精品课程的先进教学理念,每一章的开头都有学习要点和学习难点,明确本章应掌握的知识点,程序设计技能,指导读者的学习;通过具体问题分析引入相关概念的学习,精心设计大量典型例题和常用算法,讲解由浅入深,细致详尽,独特的【想一想】进一步引导学生独立思考,培养从程序设计的角度出发看问题、想问题的能力;每一章结尾都有本章小结提纲挈领,强化知识点、算法、编程方法。书中所有例题均在 Visual C++ 6.0 环境下调试通过,并提供运行结果的贴图。课后习题分选择、填空、读程序写结果、程序纠错、编程实战,类型丰富,涵盖全部关键知识点。

本书作为 C 语言程序设计省级精品课程主讲教材,语言通俗易懂、简洁流畅,内容叙述深入浅出、突出重点,不仅适合作为普通高校本、专科学生学习 C 语言的教材,也可以作为计算机编程人员、教师和一般工程技术人员的自学与参考用书,也可作为工具书以备编程时参考常用算法。

本书配套有《C 语言程序设计习题解答与上机指导》,对各章学习给出了详尽的指导,并提供了全部习题解答,还同步安排了相应的上机实践和模拟试卷。

本教材的第 1、6、9 章由梁栋编写,第 2、8 章由迟成文编写,第 3、4、5 章和附录由李新华编写,第 7、10 章由徐超编写。李龙树教授在百忙中审阅了全部书稿,在书稿的校对、

修订过程中，张辉、欧书琴、刘波、钱翔做了大量工作。在此一并向他们表示衷心的感谢。

使用本书的教师可向作者或出版社（<http://jc.cepp.com.cn>）索取免费的教学课件、全部例题、习题的源程序和配套的考试系统，也可以通过 <http://jwc.ahu.edu.cn/ec2007/C1/Course/Index.htm> 访问我们的精品课程网站，了解教学大纲、教学学时安排，进一步和作者取得联系。因编者水平有限，书中错误在所难免，恳请读者批评指正，我们会在网站上及时公布教材勘误表。

编 者

2008 年 10 月

目 录

序

前 言

第 1 章 C 程序设计概述.....	1
1.1 程序与程序设计	1
1.2 算法及其描述	3
1.3 C 语言程序的构成和格式.....	6
1.4 C 语言标识符.....	11
1.5 C 语言程序的开发.....	12
小结	17
习题	17
第 2 章 数据及运算.....	20
2.1 C 语言的数据类型.....	20
2.2 常量	21
2.3 变量	24
2.4 运算符与表达式	27
2.5 算术运算符及算术表达式	30
2.6 赋值运算符及赋值表达式	33
2.7 关系运算符及关系表达式	35
2.8 逻辑运算符及逻辑表达式	37
2.9 逗号运算符及逗号表达式	40
2.10 条件运算符与条件表达式	41
2.11 长度运算符和数据类型转换运算符.....	42
2.12 位运算符	44
2.13 数据的输入输出	47
小结	53
习题	54
第 3 章 流程控制.....	58
3.1 顺序	58
3.2 选择	63
3.3 循环	75
小结	96
习题	96
第 4 章 数组	105
4.1 顺序数据处理	105

4.2	一维数组.....	107
4.3	多维数组.....	124
4.4	字符数组及字符串处理.....	129
	小结.....	137
	习题.....	137
第 5 章	指针.....	144
5.1	指针和指针变量.....	144
5.2	指针变量的定义和使用.....	146
5.3	指针和数组.....	153
5.4	指针和字符串.....	160
5.5	指针和动态内存分配.....	163
	小结.....	166
	习题.....	167
第 6 章	函数.....	176
6.1	函数的基本概念.....	176
6.2	函数的定义、说明与调用.....	179
6.3	函数间的数据传递.....	182
6.4	函数的嵌套与递归.....	191
6.5	指针函数与函数指针.....	195
6.6	main 函数与命令行参数.....	199
6.7	局部变量和全局变量.....	201
6.8	变量的存储类型和作用域.....	204
6.9	编译预处理.....	209
	小结.....	214
	习题.....	215
第 7 章	结构体.....	230
7.1	结构体与结构体变量.....	230
7.2	结构体数组.....	240
7.3	结构体与指针.....	245
7.4	结构体与链表.....	251
7.5	共用体.....	257
7.6	枚举.....	261
7.7	类型定义符 typedef.....	263
	小结.....	264
	习题.....	265
第 8 章	文件.....	276
8.1	C 文件概述.....	276
8.2	文件的打开与关闭.....	278
8.3	文件的写读.....	281

8.4 文件的定位	291
8.5 文件的错误检测	293
小结	294
习题	294
第 9 章 C 语言实际应用	302
9.1 实例一 找最大数	302
9.2 实例二 学生成绩管理	311
9.3 实例三 数据压缩	321
小结	328
第 10 章 C 语言常见错误及解决方法	330
10.1 常见错误及解决方法	330
10.2 C 语言程序设计的基本编程规范	339
附录	345
参考文献	355

第 1 章 C 程序设计概述

【本章学习要点】

1. 了解程序和程序设计
2. 了解算法及 N-S 流程图表示法
3. 掌握 C 源程序的构成和书写规则
4. 掌握 C 语言标识符的概念
5. 掌握 C 语言基本数据输入输出的方法
6. 掌握 Visual C++ 6.0 (简称 VC++、VC) 开发环境下 C 程序的开发

【本章学习难点】

1. 模仿例题, 学会正确书写 C 源程序
2. 掌握 Visual C++ 6.0 开发环境下 C 程序的调试

1.1 程序与程序设计

1.1.1 程序

计算机是由硬件和软件两部分构成的。硬件是计算机的物质基础, 而软件则是计算机的灵魂。没有软件, 计算机就是一台裸机, 什么也干不了。计算机在安装软件, 也就是人们通常使用的程序和资料以后, 才能真正动起来, 成为一台真正的“电脑”。程序就是一组计算机指令的有序集合。通常, 人们使用计算机语言编写程序都是为了解决某个特定的问题。计算机用户之所以在自己的电脑中安装不同的应用程序就是为了解决不同的问题。

随着计算机技术及其应用的迅猛发展, 计算机被广泛地应用于人类生产、生活的各个领域。从信息的收集、整理、加工一直到传输, 人们已经越来越离不开计算机。信息时代要求人们学习和掌握编写计算机程序的基本知识和基本技能, 并把它作为自己应该具备的基本素质。

1.1.2 程序设计

程序设计就是编制程序的过程。一般要先分析待解决的问题, 提出解决问题的方法和步骤, 然后再选用一门合适的计算机程序设计语言实现上述算法。

随着计算机技术的发展, 计算机程序设计语言逐渐从机器语言、汇编语言发展到了高级语言。

1. 机器语言

机器语言是第一代计算机语言, 它的运算效率是所有语言中最高的。

计算机的大脑——CPU 所使用的指令是由“0”和“1”组成的二进制序列。用机器语言编写程序, 就是直接书写 CPU 指令。虽然计算机可以理解、执行, 但这类程序难写、难记、难理解, 而且由于各种计算机的指令系统往往各不相同, 所以在一台计算机上执行的这类程序很难在另一台计算机上顺利运行, 必须重新编写程序。

2. 汇编语言

为了减轻使用机器语言编程的麻烦，人们开始尝试用一些简洁的“助记符号”来替代由 0、1 组成的 CPU 指令，比如，用“ADD A, B”代表“ $A+B \rightarrow A$ ”，用“SUB A, B”代表“ $A-B \rightarrow A$ ”等，这样一来，不仅编写程序变得方便了，而且别人也能很容易读懂程序在干什么。

这就是第二代计算机语言——汇编语言。用汇编语言编写的程序被称之为源程序，不能直接在计算机上运行，需要一个专门的程序负责将这些“助记符号”翻译成二进制的机器目标代码才能在计算机上运行。

虽然汇编语言仍然面向计算机硬件，移植性不好，但用汇编语言编写的程序精炼，执行效率非常高，能准确发挥计算机硬件的性能和优势，所以汇编语言至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

随着人类生产、生活的各个领域对计算机程序的需求不断增多，许多接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用的第三代计算机语言——高级语言不断涌现，影响较大、使用较普遍的有 FORTRAN、BASIC、Pascal、C、C++、Visual C++、VB、Delphi、JAVA、C#等。

刚开始开发软件时，大家都是各自为战，缺乏科学规范的系统规划与测试、评估标准，这种个体手工作坊的闭门造车方式导致了大批耗费巨资开发好的软件由于含有太多的错误而无法使用，甚至带来巨大损失，这就是在计算机发展史上最早出现的“软件危机”。这时，人们认识到程序的设计应易于保证正确性，也便于验证正确性，高级语言从早期语言发展到了面向过程的结构化程序设计语言。

用高级语言编写的源程序需要将其翻译成目标代码才能在计算机上运行。完成编译任务的程序又分为两类：一类是编译一句，执行一句，被称为“解释程序”；一类是先将源程序全部翻译成目标代码以后，直接执行在磁盘上产生的目标程序，被称为“编译程序”。而解释程序不会在磁盘上产生目标程序。

对于一名身处信息时代的大学生而言，学习并掌握一门程序设计语言，培养自己运用语言编写一些小规模的应用程序解决实际问题的能力是必需的。而从中学会分析问题、解决问题，进而锻炼科学的思维方法，培养严谨踏实的科研能力尤为重要。本书正是希望通过对 C 语言的语法规则、数据类型、数据运算、语句、程序控制结构、函数的介绍和学习，培养读者掌握 C 语言程序设计的方法和技能，掌握使用 C 语言编程解决具体问题的计算机应用能力，为进一步学习程序开发奠定良好的基础。

1.1.3 C 语言的发展

C 语言是在 20 世纪 70 年代初美国贝尔实验室在开发 UNIX 操作系统时推出的程序设计工具。由于 C 语言简洁明了，使用方便、灵活，大多数操作系统和系统软件，如 Windows、UNIX、Linux 等都采用 C 语言开发，许多新的语言，如 C++、JAVA 等也由 C 语言衍生而来，因而 C 语言成为国内外程序开发人员广泛使用的程序设计工具之一。

随着 C 语言的风靡世界，各种版本的 C 编译系统层出不穷。虽然 Brian W. Kernighan 和 Dennis M. Ritchie 于 1988 年合著了著名的《The C programming Language》一书（简称《K&R》），奠定了 C 语言的基础，但是在《K&R》标准中并没有定义一个完整的 C 语言标准，后来由美

国国家标准协会 (American National Standards Institute) 在此基础上不断发展和扩充, 制定了 ANSI C 标准。1990 年国际标准化组织 (ISO) 公布了以 ANSI C 为基础制定的 ISO C, 这就是人们通常所称的标准 C。

目前在微机上广泛使用的 C 语言编译系统有 Turbo C、Turbo C++、Microsoft Visual C++ 等。它们基本上相同, 但还是存在一些差异。请读者在开发 C 程序时注意自己所使用的 C 编译系统的特点和规定。本书将以标准 C 为基础叙述 C 语言语法知识, 以目前使用最广泛的 Visual C++ 作为 C 语言程序开发平台, 书中全部例题均在 Visual C++ 6.0 上通过验证。

C 语言的主要特点可以概括如下:

(1) C 语言既可面向计算机硬件和系统, 像汇编语言一样直接访问硬件, 程序效率高; 又有高级语言面向用户, 容易记忆, 容易书写, 更容易阅读, 代码质量高的优点。

(2) C 语言程序设计是结构化程序设计。C 语言不仅具有顺序、选择、循环三种基本结构, 它还利用函数将整个程序分割成若干独立的功能模块, 为软件开发模块化和多人协同开发提供了有力的支持。

(3) C 语言数据类型十分丰富, 不仅具有字符、整数、浮点等基本数据类型, 还具有数组、指针、结构等导出数据类型, 可以根据用户需要自行定义所需的数据结构。C 语言还利用指针统一了对各种不同类型数据的访问。

(4) C 语言运算符极其丰富, 不仅具有四则运算符、关系运算符、逻辑运算符, 还具有 ++ (自增)、-- (自减)、位与、位或、移位等位运算。而括号、赋值号、逗号等符号也可以作为运算符处理, 从而使 C 语言的运算简洁、紧凑、方便、灵活。

(5) C 语言程序移植性较之汇编语言要好。C 语言本身不提供依赖于硬件的输入/输出功能, 而是通过调用统一语法但独立于 C 语言之外的编译系统函数库来实现此功能。这样便可以很容易地不同硬件架构的计算机上实现 C 程序的移植。

1.2 算法及其描述

1.2.1 算法

PASCAL 之父 Niklaus Wirth 在解释程序设计时提出了“算法+数据结构=程序”这一著名公式。程序设计先要分析待加工的数据对象, 通过在程序中指定数据的类型和数据的组织形式来描述数据对象。而算法则是程序设计中为了解决实际问题而采取的方法和步骤。

程序设计的主要任务就是分析问题, 提出解决问题的算法。学习高级语言, 一方面要熟练掌握语言的语法规则, 这是实现算法的基础, 另一方面必须训练自己的思维, 锻炼自己分析问题、分解问题, 归纳整理出算法, 最终写出高质量的程序的能力。

下面通过示例来介绍如何设计一个算法。

【例 1-1】 输入三个整数, 然后输出其中最大的数。

对于求最大数这类问题, 我们不仅要定义三个变量 first、second、third 存放从键盘输入的三个整数, 还要再准备一个变量 max 存放找到的最大数, 这便完成了本问题的数据结构设计。

从键盘输入的三个整数之后, 由于计算机一次只能对两个数进行比较, 我们可以先将 first 与 second 比, 将大的数放入 max 中, 再将 max 与 third 比, 再把大的数放入 max 中。最后,

在屏幕上输出 `max` 的值，这便完成了本问题的算法设计。

上述算法可以简化描述如下：

第一步：输入三个整数存放到变量 `first`、`second`、`third` 中。

第二步：将 `first` 与 `second` 进行比较，若 `first` 大，则将 `first` 的值放入 `max` 中，若 `second` 大，则将 `second` 的值放入 `max` 中。

第三步：将 `max` 与 `third` 进行比较，若 `third` 大，则将 `third` 的值放入 `max` 中。

第四步：在屏幕上输出 `max` 的值。

这样的算法虽然正确，但写起来很是繁琐。

进一步改进的算法如下：

S1: 输入三个整数 `first`、`second`、`third`

S2: `max = first`

S3: 若 `second > max`，则 `max = second`

S4: 若 `third > max`，则 `max = third`

S5: 输出 `max`。

这样的算法逻辑更加简洁明了，可以很方便地转化为相应的 C 语言程序。

【例 1-2】 输入四个浮点数，然后输出其中最小的数。

求最小数和求最大数实际上是同一类问题，只要对 [例 1-1] 稍作修改即可：

S1: 输入四个浮点数 `first`、`second`、`third`、`fourth`

S2: `min = first`

S3: 若 `second < min`，则 `min = second`

S4: 若 `third < min`，则 `min = third`

S5: 若 `fourth < min`，则 `min = fourth`

S6: 输出 `min`。

一个算法的优劣，可以用以下指标衡量：

(1) 可行性。算法必须是可行的，即算法所描述的操作都是可以通过执行有限次已知基本运算来实现。

(2) 有穷性。算法必须能在有穷时间内执行有穷步后完成。

(3) 确定性。算法中每一条指令都必须有确切的含义，不存在二义性。算法只有一个入口和一个出口。

1.2.2 算法的描述

描述算法的方法有很多，通常可以使用自然语言、传统流程图、N-S 流程图、PAD 流程图等。在 [例 1-1] 和 [例 1-2] 中，所使用的便是自然语言描述。虽然自然语言描述的算法通俗易懂，但算法结构很不清晰。对于一些比较复杂的问题，很难用自然语言描述清楚解决问题的方法和步骤。

传统流程图利用不同形状的几何图形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，

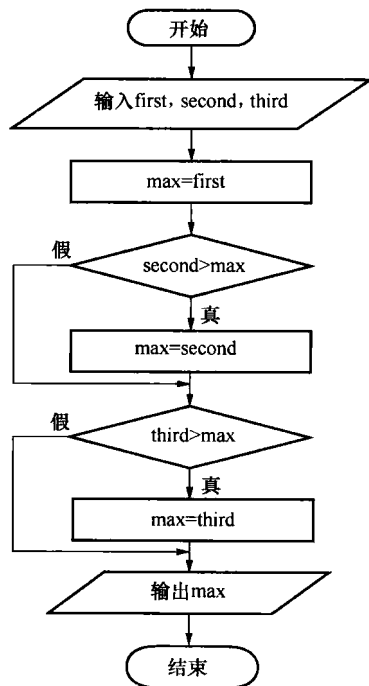


图 1-1 [例 1-1] 传统流程图的算法描述

图 1-1 为 [例 1-1] 的传统流程图算法描述。由于传统流程图简单直观,可以简洁明了地表述算法,所以在早期程序设计阶段成为程序员们交流的重要手段,得到了广泛的应用。

传统流程图的流程线允许程序从一个地方直接跳转到另一个地方去。执行这样操作的好处是程序设计十分方便灵活,减少了人工复杂度,但也使得程序的流程变得复杂紊乱,难以看懂,也难以验证程序的正确性,如果有错,排起错来更是十分困难。这种转来转去的程序设计正是导致软件危机的一个重要原因。

经过研究,人们发现,任何复杂的算法都可以由顺序结构、选择结构和循环结构这三种基本结构组成,并且基本结构之间可以并列、相互包含,但不允许交叉,不允许从一个结构直接转到另一个结构的内部去。当整个算法都由三种基本结构组成时,算法结构清晰,易于书写、阅读、实现、纠错。遵循这种方法的程序设计被称为结构化程序设计。

N-S 流程图是美国学者 I.Nassi 和 B.Shneiderman 提出的一种无流程线的结构化描述方法。N-S 图描述的算法是一个大矩形框,大框内又包含若干基本结构的矩形框,三种基本结构的 N-S 流程图描述如下所示:

1. 顺序结构

顺序结构如图 1-2 所示,它是一种最常用的线性结构,各框自上而下顺序执行。图 1-2 的执行顺序为:先 A 后 B。

2. 选择结构

选择结构如图 1-3 所示,这种结构是对给定条件 A 进行判断,当条件 A 为真时执行 B,当条件 A 为假时执行 C。

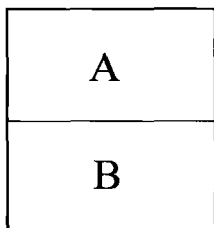


图 1-2 顺序结构的 N-S 图

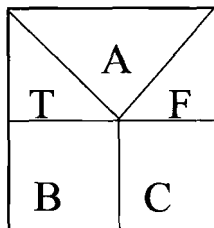


图 1-3 选择结构的 N-S 图

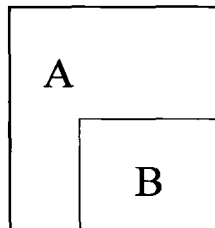


图 1-4 while 循环结构的 N-S 图

3. 循环结构

循环结构有两种基本形态: while 型循环和 do-while 型循环。

while 型循环的 N-S 图如图 1-4 所示,当条件 A 为真时反复执行循环体 B,直到条件 A 为假时才跳出循环。do-while 型循环反复执行循环体 B,直到条件 A 为假时才跳出循环。

[例 1-1] 的算法 N-S 流程图如图 1-5 所示。比较图 1-1 和图 1-5 可发现, N-S 图比流程图更加简练,更加形象直观。由于没有流程线, N-S 图比流程图更易于实现,特别适合结构化程序设计,所以推荐初学者在一开始学习程序设计时就使用 N-S 流程图来分析、研究算法。

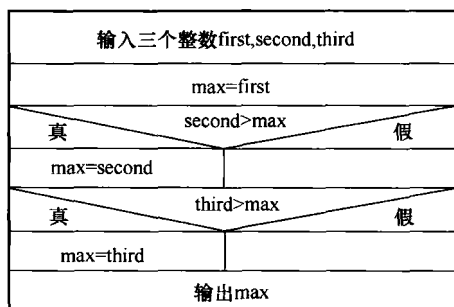


图 1-5 [例 1-1] 的算法 N-S 流程图

1.3 C 语言程序的构成和格式

学习 C 语言，首先要学习 C 语言特定的语法规则和规定的表达方法。一个严格按照 C 语言语法和表达方式的规定编写出来的 C 语言程序，不仅容易阅读、理解，还能保证编写的程序能在 Visual C++ 环境中正确地执行。

一个标准的 C 程序结构一般包括以下内容：

/* 注释 */	
#include <.....>	文件包含说明
Global declarations;	全局变量/函数说明
type functionname (type variable, ...)	函数说明
{	
local variable declarations;	函数体：局部变量说明
statements;	函数体：执行语句
}	
void main(void)	主函数说明
{	
local variable declarations;	函数体：局部变量说明
statements;	函数体：执行语句
}	

结合几个程序实例来了解 C 语言的基本程序结构。

【例 1-3】 简化 [例 1-1] 的要求：输入两个整数，然后输出其中最大的数。

```

/* Program: EG01-03.c */
/* Description: Comparing first and second, see which is the bigger. */

#include <stdio.h>                                /* 包含标准 I/O 库函数的说明 */

void main( void )                                /* 主函数说明 */
{
    int first, second, big;                       /* 局部变量说明 */

    printf("请输入两个整数:");                   /* 显示输入提示文字 */

    scanf("%d,%d", &first, &second);             /* 从键盘输入数据存入变量 */

    if( first > second )                          /* 求大数 */
        big=first;
    else
        big=second;

    printf("第一个数是 %d\n", first );            /* 输出结果 */
    printf("第二个数是 %d\n", second );
    printf("大数是 %d\n", big);
}

/* The answer is:
请输入两个整数:19,68
第一个数是 19

```


第二个数是 68

大数是 68

*/

(1) 位于程序开始的“/* */”表示程序注释,用于说明算法、标明变量的用途,帮助其他人阅读、维护该程序。注释文字可以是英文,也可以是中文。注释不会增加可执行程序的大小。编译器在编译程序时会忽略所有注释。

我们可以根据需要在程序的任意位置添加注释,比如[例 1-2]中的程序注解和末尾的运行结果。通常在要解释的代码上方或右侧放置注释。

“/* */”被称为块注释或多行注释,书写时可以跨行;但“/* */”不能嵌套,这意味着在块注释文字中不能再次出现“/*”、“*/”。初学 C 语言很容易犯这种错误。

在 Visual C++中有两种类型的注释,除了上述的块注释之外,还提供了以双斜线(//)开头的单行注释。单行注释中双斜线右侧的所有内容都是注释,它的作用范围只能到本行结束,不允许跨行。由于单行注释简洁明了,允许在注释文字中再次出现双斜线,不易出错,在现代 C 语言程序设计中得到了广泛应用。

(2)“#include <stdio.h>”(或“#include "stdio.h"”)是预处理命令中的文件包含命令,其意义是把尖括号或引号内指定的文件包含到本程序来,成为本程序的一部分。被包含的文件通常是由系统提供的,其扩展名为.h。因此也称为头文件或首部文件。C 语言的头文件中包括了编译系统提供的各个标准库函数的有关说明(函数原型)。如果在程序中调用了库函数时,就必须包含该函数原型所在的头文件。在[例 1-2]中,使用了两个库函数:标准输入函数 scanf,标准输出函数 printf,其头文件为 stdio.h,所以需要 include 命令包含了 stdio.h 文件。

(3)“void main(void){……}”是 C 语言程序定义的主函数,main 是主函数名,main 后的括号()是函数的标志,括号()中的 void 表示没有入口参数,即调用 main 函数时不需要什么已知条件,main 前的 void 表示调用完 main 函数不返回任何数据。void main(void)是主函数说明。“{……}”中包含的是函数的函数体。

(4)“int first, second, big;”定义了三个仅在 main 函数内部使用的能够存放整数的变量(int:整型变量),前两个用于存放输入的两个整数,big 用于存放即将找到的大数。

(5)“printf(“请输入两个整数:”);”调用了标准输出库函数 printf,在屏幕上显示输入提示文字。

(6)“scanf(“%d,%d”, &first,&second);”调用了标准输入库函数 scanf 从键盘输入两个整数,存入变量 first、second 对应的内存单元。输入格式字符串“%d,%d”表示要输入两个十进制整数,彼此之间用“,”分隔。取地址运算符“&”可以获得其后变量的内存单元地址。

(7)“if…else…”是选择结构,将 first 与 second 进行比较,若 first 大,则将 first 的值放入 big 中,若 second 大,则将 second 的值放入 big 中。

(8)程序最后的三个 printf 调用负责在屏幕上显示程序运行结果。输出格式字符串中的“%d”表示在这个位置上输出紧跟输出格式字符串后面的变量的十进制整数值。“\n”是换行符,将本行以后要显示的文字换到下一行显示(回车换行)。

程序的运行结果如图 1-6 所示。

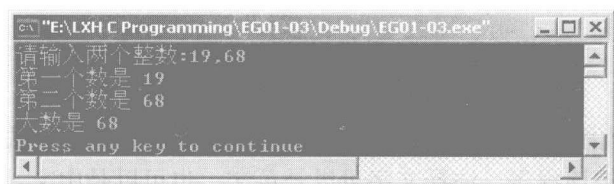


图 1-6 [例 1-3] 的程序运行结果

【例 1-4】 用 C 语言编程实现 [例 1-2]: 输入四个浮点数, 然后输出其中最小的数。

```
/* Program: EG01-04.c */
/* Description: 比较输入的四个浮点数, 打印最小数 */

#include <stdio.h>                                /* 包含标准 I/O 库函数的说明 */

void main( void )                                /* 主函数说明 */
{
    float first, second, third, fourth, min;      /* 局部变量说明 */

    printf("请输入四个浮点数:");                /* 显示输入提示文字 */

    scanf("%f, %f, %f,%f", &first, &second, &third, &fourth);
                                                /* 输入数据 */

    min = first;
    if( second < min )                            /*求 first、second 中的小数*/
        min = second;
    if( third < min )                             /*求 third、min 中的小数*/
        min = third;
    if( fourth < min )                           /*求 fourth、min 中的小数*/
        min = fourth;

    printf("%f、%f、%f、%f 中的最小数是%f \n",
           first, second, third, fourth, min); /* 输出结果 */
}
```

(1) “float first, second, third, fourth, min” 定义了五个仅在 main 函数内部使用的能够存放浮点数的变量 (float: 单精度浮点型变量), 前四个用于存放输入的四浮点数, min 用于存放即将找到的最小数。

(2) “printf(“请输入四个浮点数:”);” 调用了标准输出库函数 printf, 在屏幕上显示输入提示信息。

(3) “scanf(“%f, %f, %f,%f”, &first, &second, &third, &fourth);” 调用了标准输入库函数 scanf 从键盘输入四个浮点数, 分别存入变量 first、second、third、fourth 对应的内存单元。输入格式字符 “%f” 表示要输入一个单精度浮点数。

(4) “min = first;” 表示将 first 的值放入 min 中。

(5) “if(second < min) min = second;” 是单分支选择结构, 将 second 与 min 进行比较, 若 min 大, 则将 second 的值放入 min 中。

(6) 程序最后的 printf 调用依次从紧跟输出格式字符串后面的五个变量中取值显示在输