“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定
普通高等教育“十一五”国家级规划教材

C语言程序设计与实训

第2版

周屹 李萍◎主编

机械工业出版社
CHINA MACHINE PRESS

www.cmpedu.com
配电子课件

“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定
普通高等教育“十一五”国家级规划教材

C语言程序设计与实训

第2版

主　编　周　屹　李　萍
副主编　聂相举　钟玉峰
参　编　李　朴　王　丁
主　审　王培东

机械工业出版社

本书是“十二五”职业教育国家规划教材，经全国职业教育教材审定委员会审定。本书在第 1 版的基础上，对各章节进行了局部内容和结构的调整，完善了例题并增加大量的实训案例，深入浅出地讲解了 C 语言基本概念、数据类型、基本结构、程序设计方法及典型设计案例。相比于第 1 版，本书层次更清晰、例题更丰富、实用性更强，便于教学组织和实践操作，注重培养学生的程序设计能力。

本书可作为普通高等院校应用型本科理工类专业学生的程序设计教材，也可以作为计算机专业本、专科学生学习计算机语言的入门教材。

为方便教学，本书配备电子课件等教学资源。凡选用本书作为教材的教师均可登录机械工业出版社教育服务网 www.cmpedu.com 免费下载。如有问题请致信 cmpgaozhi@sina.com，或致电 010-88379375 联系营销人员。

图书在版编目（CIP）数据

C 语言程序设计与实训 / 周屹，李萍主编. —2 版. —北京：机械工业出版社，2016.1

“十二五”职业教育国家规划教材　普通高等教育“十一五”国家级规划教材

ISBN 978-7-111-52588-2

Ⅰ. ①C…　Ⅱ. ①周…　②李…　Ⅲ. ①C 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（2015）第 308200 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）

策划编辑：刘子峰　责任编辑：刘子峰

责任校对：张　薇　封面设计：陈　沛

责任印制：李　洋

北京振兴源印务有限公司印刷

2016 年 1 月第 2 版 · 第 1 次印刷

184mm×260mm · 16.5 印张 · 426 千字

0001－3000 册

标准书号：ISBN 978-7-111-52588-2

定价：35.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

服务咨询热线：010-88379833

读者购书热线：010-88379649

网络服务

机 工 官 网：www.cmpbook.com

机 工 官 博：weibo.com/cmp1952

教育服务网：www.cmpedu.com

金　书　网：www.golden-book.com

封面无防伪标均为盗版

前　言

在众多的程序设计语言中，C 语言以其灵活性和实用性受到了广大计算机应用、设计人员的喜爱，也一直是许多计算机专业人员和程序爱好者学习程序设计的首选入门语言。

本书第 1 版于 2008 年 2 月出版，是普通高等教育“十一五”国家级规划教材。为了适应发展需要，并结合这些年的教学实践以及广大读者提出的建议和需求，我们对原书从结构到内容做了较大的增删和修改，特别是各章节增加了例题，方便教师教学，同时让学生更好地掌握和理解 C 语言，全面培养程序设计思想和能力。

本书由两部分构成：第一部分是 C 语言基础，第二部分是案例实训。课堂讲授时可根据学生及专业情况对内容酌情取舍。由于学时的限制，本书的第二部分可作为学生课外练习或课程设计内容，同时可对学生做不同层次的要求。

在内容安排上，本书在考虑与一般教材兼容的同时，在实用性方面又做了补充。与第 1 版相比，本次修订主要有以下几方面的调整：第 1 章中增加了算法与结构化程序设计的内容；调整了第 5～8 章整体框架和各小节的层次结构顺序，并对内容进行了调整；第 14 章中增加了简单计算器的内容；增加了第 15 章综合案例设计——游戏和第 16 章 C++简介，删除原第 15 章 Windows 程序设计；每章增加了许多连续性的例题，实训部分增加了实训目的和丰富的案例内容。

本书由长期承担程序设计基础课教学、具有丰富编程经验的一线教师编写。作者根据多年从事程序设计课程教学活动以及软件开发的经验，针对学生的认知规律，在详细阐述 C 语言基础知识的基础上，着重讨论了程序设计的基本原理、概念和方法，穿插演示性案例于理论讲解之中，使枯燥的理论变得更易于理解和接受。此外，每章安排了实训内容，目的是提高学生综合利用所学知识解决实际问题的能力。

本书由周屹、李萍主编，聂相举、钟玉峰任副主编，参加编写的还有李朴、王丁。其中，第 1、2、6、7、9 章由周屹编写，第 3、4、8、10 章由李萍编写，第 11、15 章由聂相举编写，第 5、12、16 章由钟玉峰编写，第 13 章和附录由李朴编写，第 14 章由王丁编写，全书由周屹统稿。王培东教授任主审。

本书在编写过程中，得到了各方面有关专家的大力支持和帮助，在此对所有人的工作与支持表示衷心的感谢。由于编者水平有限，书中难免存在不足，敬请广大读者批评指正。

编　者

目　录

第一部分 C语言基础

第1章 C语言概述

C语言具有简洁紧凑、灵活方便、运算符与数据结构丰富、生成代码质量高、程序执行效率高等诸多优点，成为世界上应用最广泛的几种计算机语言之一。使用它不仅可以编写系统软件，也可以编写应用软件。

1.1 C语言的演变

C语言是由UNIX的研制者Dennis Ritchie和Ken Thompson设计的。C语言是在B语言的基础上发展起来的，它的根源可以追溯到ALGOL 60。1960年出现的ALGOL 60是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序，因此在1963年英国的剑桥大学推出了CPL（Combined Programming Language）语言。CPL语言在ALGOL 60的基础上接近硬件一些，但规模比较大，难以实现。

C语言的第一次发展在1969年到1973年之间，是由一种早期的被称为B语言的编程语言发展而来的。1967年英国剑桥大学的Matin Richards对CPL语言作了简化，推出了BCPL（Basic Combined Programming Language）语言。1970年，美国贝尔实验室的Ken Thompson以BCPL语言为基础，又作了进一步简化，使得BCPL能压缩在8KB内存中运行，这个简单且很接近硬件的语言就是B语言（取BCPL的第一个字母），使用它写了第一个UNIX操作系统，并在DEC PDP-7上实现。但B语言过于简单，并且功能有限，和BCPL都是“无类型”的语言。

1973年，C语言被用于编写UNIX操作系统的内核，这是第一次用C语言来编写操作系统的内核。Dennis Ritchie和Brian Kernighan在1978年出版了《C程序设计语言》（The C Programming Language，经常简称为“白皮书”或“K&R”）。1980年以后，贝尔实验室使C语言更为广泛地流行，并一度成为操作系统和应用程序编程的首选。甚至到今天，它仍被用于编写操作系统以及作为广泛的计算机教育的语言。

随着微型计算机的日益普及，出现了许多C语言版本。1983年，美国国家标准委员会（ANSI）对C语言进行了标准化，并颁布了第一个C语言标准草案（83 ANSI C），后来于1987年又颁布了另一个C语言标准草案（87 ANSI C）。最新的C语言标准是在1999年颁布并在2000年3月被ANSI采用的C99。20世纪80年代晚期，Bjarne Stroustrup和贝尔实验室为C语言添加了面向对象的特性，扩展出C++语言。目前，C++语言已广泛应用于基于Microsoft Windows系统平台的商业应用程序的编写。然而C语言的特点和优势，使其仍然是UNIX世界的热门编程语言。

1.2 C语言的特点

C语言广泛应用于不同的操作系统，如UNIX、MS-DOS、Microsoft Windows及Linux

等。C 语言是一种面向过程的语言，同时具有高级语言和汇编语言的优点。在 C 语言的基础上发展起来的有支持多种程序设计风格的 C++语言，以及现在网络上广泛使用的 Java、JavaScript、微软的 C#等。

许多著名的系统软件，如 DBASE Ⅲ PLUS、DBASE Ⅳ、Windows 操作系统、UNIX 操作系统等都是由 C 语言编写的。使用 C 语言并加上一些汇编语言的子程序，就更能显示出 C 语言的优势，如 PC-DOS、WORDSTAR 等。归纳起来，C 语言具有下列特点。

1．C 语言是结构化语言

结构化语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护及调试。C 语言是以函数形式提供给用户的，这些函数可方便地调用，提供了多种循环语句和条件语句实现控制程序流向，使程序完全结构化。

2．C 语言功能齐全

C 语言具有各种各样的数据类型，并引入了指针概念，使程序效率更高。C 语言还具有强大的图形功能，支持多种显示器和驱动器，而且计算功能、逻辑判断功能也比较强大，可以实现决策目的。

3．C 语言适用范围广

C 语言的一个突出优点就是适用于多种操作系统，同时也适用于多种机型。由于实现了对硬件的编程操作，因此 C 语言集高级语言和低级语言的功能于一体，既可用于系统软件的开发，也适合于应用软件的开发。

C 语言还具有生成目标代码质量高、程序执行效率高的特点。此外，与汇编语言相比，用 C 语言编写的程序可移植性好。但 C 语言对程序员要求也高，较其他高级语言在学习上要困难一些。

目前较为流行的 C 语言有几个版本：Microsoft C 或称为 MS C、Borland Turbo C 或称为 Turbo C、AT&T C，这些 C 语言版本不仅实现了 ANSI C 标准语法，而且在此基础上各自作了一些扩充，使之更加方便、完美。

4．C 语言的语法特点

1）C 语言简洁、紧凑，使用方便、灵活。ANSI C 一共有 32 个关键字，在 C 语言中，关键字都是小写的。

2）运算符丰富，共有 34 种。C 语言把括号、赋值符号、逗号等都作为运算符处理。C 语言丰富的运算符，可以实现其他高级语言难以实现的运算。

3）数据结构类型丰富。

4）具有结构化的控制语句。

5）语法限制不太严格，程序设计自由度大。

6）C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。

在 C 语言的基础上，贝尔实验室又推出了 C++语言。C++语言进一步扩充和完善了 C 语言，成为一种面向对象的程序设计语言。C++语言目前流行的版本有 Borland C++ Builder 6.0、Microsoft Visual C++ 6.1 和 Microsoft Visual C++ 2012。C 语言是 C++语言的基础，C++语言和 C 语言在很多方面是兼容的，因此掌握了 C 语言，再进一步学习 C++语言，以一种熟悉的语法来学习面向对象的语言，可以达到事半功倍的效果。

1.3　C 语言的编写过程

程序是为实现特定目标或解决特定问题而用计算机语言编写的命令序列集合，即使用某种计算机语言编写程序来完成特定任务。用 C 语言编写的程序称为源程序（Source Program）。C 语言程序设计是一种结构化程序设计，它利用 C 语言运行环境来实现各种算法，以完成特定的目标。

1.3.1　程序开发过程

将源程序翻译成机器语言的过程称为编译，由于计算机本身并不能直接理解这样的程序，需要经过解释程序或编译程序将其翻译成机器语言，计算机才能理解并运行程序。编译的结果是得到了源程序的目标代码（Object Program），还要将目标代码与系统提供的函数和自定义的过程（或函数）链接起来，最后得到机器可执行的程序，称为可执行程序或可执行文件。

1．源程序翻译

C 语言源程序的扩展名为.c。它是不能直接在计算机上运行的，必须通过机器翻译成目标代码，再将目标代码链接成可加载模块（可执行文件），才能在计算机上运行。这种把源程序翻译成目标代码的程序称为编译器或翻译器。适合 C 语言的编译器不止一种，不同的机器、不同的操作系统可能会有一种或多种不同的编译器。C 语言源程序的翻译过程如图 1-1 所示。它由词法分析器、语法分析器和代码生成器 3 部分组成。

图 1-1　程序语言翻译过程

（1）词法分析器

词法分析器（Lexical Analyzer）主要是对源程序进行词法分析，它是按单个字符的方式阅读源程序，并且识别出哪些符号的组合可以代表单一的单元，并根据它们是否是数字值、单词（标识符）、运算符等，将这些单词分类。词法分析器将词法分析结果保存在一个结构单元里，这个结构单元称为记号（Token），并将这个记号交给语法分析器。词法分析器会忽略源程序中的所有注释。

（2）语法分析器

语法分析器（Parser）直接对记号进行分析，并识别每个成分所扮演的角色。这些语法规则也就是程序设计语言的语法规则。

（3）代码生成器

代码生成器（Code Generator）将经过语法分析后没有语法错误的程序指令转换成机器语言指令。

如果源程序没有错误，就会生成一个目标代码程序，具体的命令与每种程序语言的编译器有关。

2．链接目标程序

通过翻译产生的目标代码程序尽管是机器语言的形式，但却不是机器可以执行的方式，目标程序只是一些松散的机器语言，要获得可执行的程序，还需要将它们链接起来。

程序的链接工作由链接器（Linker）来完成。链接器的任务就是将目标程序链接成可执行的程序（或称载入模块），这种可执行的程序是一种可存储在磁盘存储器上的文件。

上面对程序进行编译、链接都只针对了一个源程序文件，实际上，可以将多个源程序文件通过编译，链接成一个可执行文件。

对于程序的编译、链接，有必要强调以下几点：

1）并不是任何目标程序都可以链接成可执行程序。

2）被链接成可执行程序的目标程序中，只允许在一个程序中有且仅有一个可被加载的入口点。

3）对于具体的程序语言，编译、链接程序的方法会有所不同，针对某一种程序语言的编译器，不可以用于对另一种源程序语言的编译。

4）上面对 C 语言进行编译、链接的方式并不是唯一的，它允许有一些其他的变化，具体可参考各编译器的使用说明。

一个 C 语言程序的具体开发步骤如图 1-2 所示。

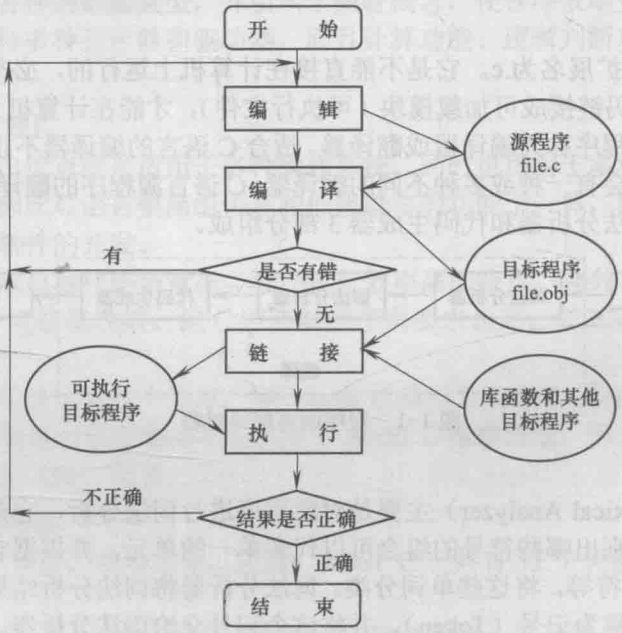

图 1-2　C 语言程序的具体开发步骤

从图 1-2 中可以看出，一个 C 语言程序的开发需经过编辑、编译、链接和执行 4 个步骤。源程序经过编译器（编译程序）被翻译成目标程序，再通过链接器（链接程序）与其他目标程序或库函数链接起来，生成可存储在外存中的可执行程序。生成了可执行程序，就可以反复地被加载执行，而不需要重新编译、链接，如果修改了源程序，也不会影响到已生成的可执行程序，除非对修改后的源程序重新编译和链接，生成一个新的可执行程序。目前 C 语言程序的开发都是在集成开发环境（如 Turbo C、Visual C++、C-Free、Dev-C++、Codeblocks 等）中进行的，程序的编辑、编译、链接和执行 4 个步骤都在一个窗口中进行，极大地简化了编程过程，提高了编程效率。

1.3.2　算法与结构化程序设计

结构化程序设计方法是从 20 世纪 70 年代开始逐步形成的一种程序设计方法。它要求程

序员按照一定的规范，采用成熟的设计方法进行程序设计。它强调程序的风格、程序结构的规范化以及自顶向下、逐步细化和模块化的设计方法，追求的主要目标是提高程序的易读性和易维护性。

Bohn 和 Jacopini 于 1966 年就提出了结构化程序设计的理念。结构化程序设计思想和方法的引入，使程序结构更清晰，易于阅读、修改和验证，从而提高了程序设计的质量和效率。

结构化程序设计方法适用高级语言表示的结构化算法。该方法的基本思路是将一个复杂问题的求解过程分阶段进行，每个阶段处理的问题都控制在人们容易理解和处理的范围内。

1．结构化程序设计的原则

1）自顶向下。程序设计时，应该先总体，后细节；先全局，后局部。一开始不要过多地追求细节，应从最上层总体目标开始，逐步使问题具体化。

2）逐步细化。复杂问题分解成一些子问题，逐步细化。

3）模块化设计。所谓模块化设计，就是按模块组装的方法编程。把一个待开发的软件分解成若干个小的、简单的部分，称为模块。每个模块都独立地开发、测试，最后再组装出整个软件。这种开发方法是在软件开发领域中对待复杂事务所采用的“分而治之”的一般原则。

模块化澄清和规范了软件中各部分间的界面，便于的软件设计人员团队工作，也促进了更可靠的软件设计实践。

4）结构化编码。软件开发的最终目的是产生能在计算机上执行的程序，即使用选定的程序设计语言，把模块的过程性描述翻译为用该语言书写的源程序（源代码）。重要的是结构化编码的思想，具备了该思想，语言就只是工具了。

总体来说，程序设计应该强调简单而清晰。结构化程序设计方法以算法为核心，程序=算法+数据结构，算法和数据结构是两个独立的整体。

早在计算机发明以前，算法就是数学家的工具，被用来描述特定问题的解决方法。开发出来的算法是以人能理解的语言描述的，为了让计算机能接受算法，计算机必须具有自己的语言系统。计算机能理解的语言被称作程序设计语言，程序设计语言规定了书写程序可使用的一组记号和一组语法规则。做任何事情都有一定的步骤，为解决一个问题而采取的方法和步骤，称为算法。计算机能够执行的算法就是计算机算法。计算机算法可分为数值运算算法和非数值运算算法两大类。

例 1-1　求 5！（5！=1×2×3×4×5）。

最开始算法为：步骤 1：先求 1×2，得到结果 2。

步骤 2：将步骤 1 得到的乘积 2 乘以 3，得到结果 6。

步骤 3：将 6 再乘以 4，得 24。

步骤 4：将 24 再乘以 5，得 120。

这样的算法虽然正确，但太繁。

改进后的算法如下：S1：使 t=1。

S2：使 i=2。

S3：使 t×i，乘积仍然放在变量 t 中，可表示为 t×i→t。

S4：使 i 的值+1，即 i+1→i。

S5：如果 i≤5，返回重新执行步骤 S3 以及其后的 S4 和 S5；否则，算法结束。

如果计算 66！只需将 S5 中的 i≤5 改成 i≤66 即可。改进后的算法不仅正确，而且是较

好的算法，因为计算机是高速运算的自动机器，实现循环轻而易举。

2．算法的特性

并不是所有组合起来的操作步骤都可以称为算法。算法必须符合以下 5 项基本特性：

1）有穷性。一个算法应包含有限的操作步骤而不能是无限的。

2）确定性。算法中每一个步骤应当是确定的，并且在任何条件下算法都只能有一条可执行路径，无歧义性。

3）可行性。算法中所有操作都必须是可执行的。如果按照算法逐步去做，将得到确定的结果。

4）有零个或多个输入。在程序运行过程中，有的数据是需要在算法执行过程中输入的，而有的算法表面上看没有输入，但实际上数据已经被嵌入其中了。没有输入的算法是缺少灵活性的。

5）有一个或多个输出。算法进行信息加工后应该得到至少一个结果，没有输出的算法是没有用的。

3．算法的表示

表示一个算法可以用不同的方法，常用的有自然语言、传统流程图、N-S 流程图、伪代码等。

（1）用自然语言表示算法

自然语言就是人们日常使用的语言，可以是汉语、英语或其他语言。用自然语言表示算法通俗易懂，但文字冗长，容易出现“歧义性”。自然语言表示的含义往往不严格，要根据上下文才能判断其正确含义，描述包含分支和循环的算法时也不方便。因此，除了有些很简单的问题外，一般不用自然语言描述算法。如例 1-1 的算法就是用自然语言表示的。

（2）用传统流程图表示算法

美国国家标准化协会（American National Standard Institute, ANSI）规定了一些常用的流程图符号，如图 1-3 所示。

例 1-1 用传统流程图表示如图 1-4 所示。

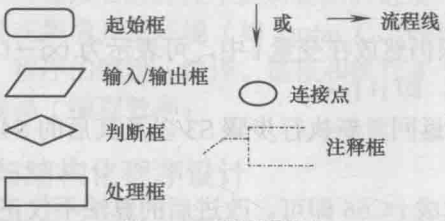

图 1-3　流程图符号

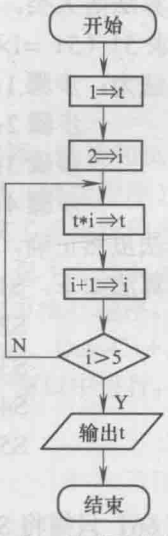

图 1-4　求 5!的算法传统流程图

传统流程图用流程线标出各框的执行顺序，对流程线的使用没有严格限制。因此，使用者可以不受限制地将流程线随意地转向，使流程图变得毫无规律，而阅读者要花大量的精力去追踪流程，使算法的逻辑难以理解。

（3）用N-S流程图表示算法

1973 年提出了一种新的流程图形式。在这种流程图中，完全去掉了带箭头的流程线，全部算法写在一个矩形框内，在该框内还可以包含其他的从属于它的框，或者说由一些基本的框组成一个大的框。这种流程图又称 N-S 结构化流程图。

用 N-S 流程图描述例 1-1 求 5! 的算法，如图 1-5 所示。

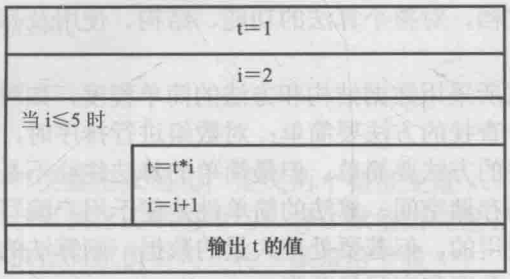

图 1-5　N-S 流程图

（4）用伪代码表示算法

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。它如同一篇文章一样，自上而下地写下来，每一行（或几行）表示一个基本操作。它不用图形符号，因此书写方便、格式紧凑，也易懂，便于向计算机语言算法（即程序）过渡，适用于设计过程中需要反复修改时的流程描述。

例 1-2　打印 x 的绝对值。

用伪代码表示为：

```
IF x is positive
THEN print x
ELSE print -x
```

也可以用汉字伪代码表示为：

```
若 x 为正　打印 x
否则　打印-x
```

还可以中英文混用，例如：

```
IF x 为正　print x
ELSE print -x
```

4. 算法评价

算法评价的目的首先是从解决同一个问题的不同算法中选择出较为合适的一种，其次是知道怎样对现有算法进行改进，进而设计出更好的算法。对于算法的评价可以从以下几个方面进行：

（1）正确性

正确性是设计和评价算法的首要条件。一个正确的算法是指在有合理的数据输入情况下，能够在有限的运行时间内得出正确的结果。要从理论上证明一个算法的正确性，并不是一件容易的事，所以通常可采用各种典型的输入数据上机调试算法，并使算法中的每段代码都通过测试，若发现错误及时修正，最终可以验证出算法的正确性。

（2）健壮性

健壮性是指一个算法对不合理（又称为不正确、非法、错误等）数据输入的反应和处理能力。一个好的算法应该能够识别出错误数据并进行相应的处理。对错误数据的处理一般包括打印错误信息、调用错误处理程序、返回标识错误的特定信息、中止程序运行等方法。

（3）可读性

可读性是指一个算法供人们阅读和理解的容易程度。一个可读性好的算法，应该符合结构化和模块化程序设计的思想；应该对其中的每个功能模块、重要数据类型或语句加以必要注释；应该建立相应的文档，对整个算法的功能、结构、使用及有关事项进行说明。

（4）简单性

简单性是指一个算法所采用数据结构和方法的简单程度，如对数组进行查找时，采用顺序查找的方法比采用二分查找的方法要简单；对数组进行排序时，采用简单选择排序的方法比采用堆排序或快速排序的方法要简单。但最简单的算法往往不是最有效的，即有可能占用较长的运行时间和较多的存储空间。算法的简单性是便于用户编写、分析和调试，所以对于处理少量数据的情况是适用的，但若要处理大量的数据，则算法的有效性比简单性更重要。有效性主要表现为时间复杂度和空间复杂度。

（5）时间复杂度

时间复杂度是指计算机执行一个算法时在时间上的消耗度量。度量一个程序的执行时间通常有两种方法：事后统计和事前分析估算。

（6）空间复杂度

空间复杂度是指在一个算法的运行过程中，对临时耗费的存储空间的度量，而不包括问题的原始数据占用的空间（因为这些单元与算法无关）。

目前由于计算机硬件的发展，一般都有足够的内存空间，因此在算法评价中应着重考虑时间因素。

1.4 简单 C 程序介绍

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由简到难，表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中了解到组成一个 C 语言源程序的基本组成和书写格式。

例 1-3 在屏幕上显示“Hello，C!”。

```
#include <stdio.h>
main()                    /*主函数*/
{
    printf("Hello, C!\n");          /*输出语句*/
}
```

C 语言源程序的结构特点如下：一个 C 语言源程序可以由一个或多个源文件组成；每个源文件可由一个或多个函数组成；一个源程序不论由多少个文件组成，都有一个且只能有一个 main 函数，即主函数。

例 1-3 中 main 是主函数的函数名，表示这是一个主函数。每一个 C 语言源程序都必须有且只能有一个主函数。一对花括号“{}”是主函数的定界符，例 1-3 中主函数中只有一个

语句，该语句中调用了格式输出库函数 printf()，用于向屏幕上输出一个字符串。printf 函数的功能是把要输出的内容送到标准输出设备上，默认的标准输出设备是显示器。由“/*”和“*/”括起来的任何文字是注释，程序不执行注释部分。语句用分号结束。

主函数体分为两部分：一部分为说明部分；另一部分为执行部分。说明是指变量的类型说明。例 1-3 中未使用任何变量，因此无说明部分。

C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点。说明部分是 C 语言源程序结构中很重要的组成部分。

例 1-4　输出两个变量的和。

```
#include <stdio.h>
void main()
{
    int x,y;          /*变量定义语句：定义两个整型变量 x、y*/
    x=2;              /*可执行的赋值语句：将 2 赋值给变量 x*/
    y=5;      /*可执行的赋值语句：将 5 赋值给变量 y*/
    printf("%d\n",x+y);
}
```

本例中使用了两个变量 x、y，用类型说明符 int 来说明这两个变量。说明部分后的 3 行为执行部分或称为执行语句部分，用以完成程序的功能。执行部分的第一、二行是赋值语句，printf 函数在显示器上输出表达式 x+y 的值 7，至此程序结束。

例 1-5　利用函数实现输出最小值。

```
#include <stdio.h>
int min(int a,int b);                /*函数说明*/
void main()                          /*主函数*/
{
    int x,y,z;                       /*变量说明*/
    printf("input two numbers：\n");
    scanf("%d%d",&x,&y);             /*输入 x、y 的值*/
    z=min(x,y);                      /*调用 min 函数*/
    printf("nin number=%d",z);       /*输出*/
}
int min(int a,int b)                 /*定义 min 函数*/
{
    if(a<b) return a; else return b; /*把结果返回主函数*/
}
```

例 1-5 中程序的功能是由用户输入两个整数，程序执行后输出其中较小的数。程序由两个函数组成，主函数 main()和求最小值函数 min()。函数之间是并列关系，可从主函数中调用其他函数。min 函数的功能是比较两个数，然后把较小的数返回给主函数。min 函数是一个用户自定义函数，因此在主函数中要给出说明。可见，在程序的说明部分中，不仅可以有变量说明，还可以有函数说明。关于函数的详细内容将在以后章节中介绍。

例 1-5 中程序的执行过程如下：首先在屏幕上显示提示串，请用户输入两个数，回车后

由 scanf 函数接收这两个数并送入变量 x、y 中；然后调用 min 函数，并把 x、y 的值传送给 min 函数的参数 a、b；在 min 函数中比较 a、b 的大小，把较小的数返回给主函数的变量 z，最后在屏幕上输出 z 的值。

C 语言词汇分为 6 类：标识符、关键字、运算符、分隔符、常量、注释符。

1. 标识符

在 C 语言中，标识符是程序的基本语法单位。标识符是一个名称，在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。

C 语言规定，标识符只能是由字母（A～Z 和 a～z）、数字（0～9）和下画线（_）组成的字符串，并且其第一个字符必须是字母或下画线。

由用户根据需要定义的标识符称为用户标识符。用户标识符一般被用来给变量、函数、数组、语句标号或文件等命名。例如，a、b、_2x、y、BOOK1、max5、yes 等均是合法的用户标识符；而-3x（以负号开头）、123$（以数字开头）、s*T（出现非法字符*）、char（系统关键字）等均是非法的用户标识符。

标识符命名时应注意以下几点：

1）“见名知意”。标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便阅读理解。所谓“见名知意”就是指通过变量名就知道变量值的含义。通常应选择能表示数据含义的英文单词（或缩写）作变量名，或汉语拼音字头作变量名，如 sum（求和）、max（最大）、PI（π）、BT（β）等。

2）大小写有区别。习惯上变量名、函数名等用小写，如 x、book、min()等；符号常量用大写，如 BOOK、PI、BT 等。其中，小写的 book 和大写的 BOOK 是两个不同的标识符。

3）用户标识符决不能与系统规定的关键字同名，如“int=3”“float=3.0”是非法的；也不要使用系统提供的标准库函数的函数名（如 printf、getchar、fabs 等）和编译预处理命令（如 define、include 等）。

4）尽量不要用下画线开头且长度不超过 8 位。原因是 Turbo C 系统内部使用了部分用下画线开头的标识符，如_fd、_cs、_ds、_ss 等十几个扩展关键字。

标准 C 语言不限制标识符的长度，但它受各种版本的 C 语言编译系统限制，同时也受到具体机器的限制。例如，在某版本 C 语言中规定标识符前 8 位有效，当两个标识符前 8 位相同时，则被认为是同一个标识符。

2. 关键字

关键字是由 C 语言规定的具有特定意义的字符串，又称为保留字。它是由系统提供的，表示特定的语法成分。

所有的关键字均为小写字母，专供系统本身使用，用户定义的标识符不应与关键字相同。

下面分别给出 ANSI C 规定的 32 个关键字，见表 1-1。

表 1-1　关键字

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

C 语言的关键字分为以下几类：