KH 科海图书
25年·IT技术出版专家

21世纪高职高专计算机技能与应用系列规划教材

C语言程序设计实训教程

李洪洋 主 编
侯 枫 高喜斌 副主编

中国人民大学出版社
北京科海电子出版社
www.khp.com.cn

21世纪高职高专计算机技能与应用系列规划教材

C语言程序设计实训教程

李洪洋　主　编
侯　枫　高喜斌　副主编

中国人民大学出版社
·北京·

北京科海电子出版社
www.khp.com.cn

图书在版编目(CIP)数据

C语言程序设计实训教程/李洪洋主编.
北京：中国人民大学出版社，2009
（21世纪高职高专计算机技能与应用系列规划教材）
ISBN 978-7-300-10152-1

Ⅰ.C…
Ⅱ.李…
Ⅲ.C语言—程序设计—高等学校：技术学校—教材
Ⅳ.TP312

中国版本图书馆CIP数据核字（2008）第204435号

21世纪高职高专计算机技能与应用系列规划教材
C语言程序设计实训教程
李洪洋　主编

出版发行	中国人民大学出版社　北京科海电子出版社		
社　　址	北京中关村大街31号	**邮政编码**	100080
	北京市海淀区上地七街国际创业园2号楼14层	**邮政编码**	100085
电　　话	（010）82896442　62630320		
网　　址	http://www.crup.com.cn		
	http://www.khp.com.cn（科海图书服务网站）		
经　　销	新华书店		
印　　刷	北京市鑫山源印刷有限公司		
规　　格	185 mm×260 mm　16开本	**版　　次**	2009年2月第1版
印　　张	18.75	**印　　次**	2009年2月第1次印刷
字　　数	456 000	**定　　价**	29.00元

内容提要

本书共分为 13 章。在系统地介绍了 C 语言的基础知识，基本数据类型、运算符和表达式，数据的输入输出，控制结构、数组、函数、编译预处理、结构体与共用体、指针、位运算和文件存储类别等知识的同时，辅以课后习题、上机操作指导和两个来自工作中的精选实例，以及在编程中的常见错误，方便读者掌握 C 语言的精髓，又不会感觉过于晦涩难懂，快速提升 C 语言程序设计技能。

本书结构清晰，内容翔实，示例丰富，通俗易懂，所精选的实例贴近工作，可操作性强，便于读者掌握并应用到实际工作中去，尤其适合作为各类高职高专院校、计算机培训学校等相关专业的教材，也可以作为程序设计爱好者的参考用书。

为方便读者学习和参考，书中全部实例和习题的源代码可到 http://www.khp.com.cn 中下载。用书教师可致电（010）82896438 或发 E-mail：feedback@khp.com.cn 免费获取电子教案。

前　言

C 语言的功能非常强大。它使用灵活，可移植性好，既具有高级语言的优点，又可以实现低级语言的许多功能；既可以编写系统软件，也可以编写应用软件，因此，C 语言是绝大部分程序设计人员和计算机爱好者学习程序设计的首选语言。

本书的目的是使教师和学生轻松、愉快地完成 C 语言程序设计课程的教学和学习，使读者不仅掌握一门编程语言，同时也掌握一种编程思想。全书采用循序渐进的方法推进讲解，由浅入深地对关键知识点进行分析，并辅以大量具有代表性和实用性的案例，使学生在掌握 C 语言程序设计的同时，也为以后学习数据结构、编译原理等课程打下良好的基础。

本书导读

1. 在学习本书以前，大部分学生应该已经上过《计算机基础》这门课。为了更好地与这门课程进行衔接，在本书的第 1 章开头部分回顾了计算机的发展以及计算机程序设计语言的历史。在第 1 章后半部分进入重点，开始介绍 C 语言的特点、C 语言的运行环境以及如何编写一个简单的 C 语言程序，引导学生进入 C 语言程序设计的学习。

2. 在本书的第 2 章和第 3 章介绍了 C 语言的基本数据类型、运算符与表达式、数据的输入输出等基础知识。在这些基础知识的基础上，从第 4 章到第 11 章介绍了 C 语言的应用，如控制结构、数组、函数、编译预处理、结构体与共用体、指针、位运算和文件等。其中，函数的使用最能代表 C 语言结构化的程序设计思想，是学习的重点之一。指针是 C 语言的精华，是 C 语言程序设计的典型特性之一，也是学习的难点，正是指针使 C 语言既具有高级语言的特点，又可以完成低级语言的功能。

3. 本书第 12 章是上机操作指导。该章基于 Visual C++ 6.0 平台，对书中必须掌握的知识都要求学生上机操作，在实践中轻松掌握。这些上机指导有助于提高学生在实际应用中的编程能力，增强学生学习 C 语言编程的信心。

4. 本书第 13 章对 C 语言程序设计方法进行了总结，并通过深入剖析两个来自实际工作中的精选实例，帮助读者综合运用所学知识，提升程序设计技能。

5. 本书最后的附录是 C 语言编程中常见错误的总结。该总结有助于促进学生对所学知识有更深刻的理解。

本书特点

1. 力求与 C++兼容。程序设计语言的发展方向为由面向过程到面向对象，而面向对象的 C++语言是这个发展趋势的最有力的代表。本书从语法和程序结构两方面都力求与 C++兼容，并且所有例题均在 Visual C++ 6.0 上调试通过。

2. 易教易学。本书在内容编排上按照循序渐进、由浅入深的原则，突出重点内容，并运用文字、图表、小实例等相结合的方式，将抽象概念形象化，便于读者学习和掌握。

3. 内容新颖富有启发性。本教材在讲解程序设计的例题时，不是简单“教”学生用程序解决问题的方法，而是启发学生去分析解决问题的过程，从中找出可以归结为规则操作的方法，进而编写程序。在实例部分对已有的方法提出改进的可能，启发有能力的同学深入学习，旨在培养学生勤于思考的习惯。

4. 注重培养学生的应用能力。大量的实例是本书的一大特色，本书通过设置独立的上机操作指导以及综合应用实例来训练、提高学生在实际工作中的应用能力。

为方便读者学习和参考，书中全部实例和习题的源代码可到 http://www.khp.com.cn 中下载。用书教师可致电（010）82896438 或发 E-mail：feedback@khp.com.cn 免费获取电子教案。

本书结构清晰，内容详实，示例丰富，通俗易懂，所精选的实例贴近工作，可操作性强，便于读者掌握并应用到实际工作中去，尤其适合作为各类高职高专院校、计算机培训学校等相关专业的教材，也可以作为程序设计爱好者的参考用书。

本书由刘洪洋主编，侯枫、高喜斌为副主编，桂君莉、张振华、赵丽萍等也为本书做了大量的工作，在此表示感谢！由于时间仓促，加上编者水平有限，书中不足之处在所难免，敬请专家和读者指正。

编者

2009 年 1 月

目　录

第 1 章

C 语言概述

本章导读

本章介绍了计算机程序设计语言的发展，以及程序的编辑、编译及运行。熟悉 C 语言程序的上机环境，是学习 C 语言程序设计的基础。

教学目标

- 计算机语言的发展历史
- C 语言的特点
- C 程序的运行环境
- C 程序编辑、编译、连接和运行的步骤

1.1 计算机程序设计基础

1.1.1 计算机的发展

自从 1946 年 2 月 14 日，世界上第一台真正意义上的计算机 ENIAC 诞生于美国的宾夕法尼亚大学以来，它已经走过了六十余年的历程。计算机按其采用的物理器件分为四代，分别是采用电子管的第一代，晶体管的第二代，中小规模集成电路的第三代和大规模、超大规模集成电路的第四代。目前，第五代计算机——智能计算机正在探索、研制阶段。

虽然计算机已经发展了四代，但其基本工作原理依然是 ENIAC 采用的冯·诺依曼原理，其基本思想是：存储程序与程序控制。存储程序是指人们必须事先把计算机的执行步骤序列（即程序）及运行中所需的数据，通过一定方式输入并存储在计算机的存储器中。程序控制是指计算机运行时能自动地逐一取出程序中一条条指令，加以分析并执行规定的操作。自计算机诞生的那一天起，这一原理就决定了人们使用计算机的主要方式——编写程序和运行程序。科学家们一直致力于提高程序设计的自动化水平，改进用户的操作界面，提供各种开发工具、环境与平台，其目的都是为了让人们更加方便地使用计算机，可以少编程甚至不编程来使用计算机，因为计算机编程毕竟是一项复杂的脑力劳动。但不管用户的开发与使用界面如何演变，“存储程序原理”没有变，它仍然是我们理解计算机系统功能与特征的基础。

1.1.2 计算机语言的发展历史

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的过程。

1. 机器语言

从某方面来说，计算机是一个相当死板的装置。由于电子元器件只能识别“通电”和“断电”两种状态，因此计算机也只能识别“0”和“1”这两种信号，也就是二进制数。计算机发明之初，人们用计算机的语言去命令计算机干这干那，也就是写出一连串的由“0”和“1”组成的指令序列交由计算机执行，这种语言，就是机器语言。然而使用机器语言是相当痛苦的，程序员要记住大量的二进制编码，特别是在程序有错需要修改时，更是如此。而且，由于每台计算机的指令系统往往不同，所以在一台计算机上执行的程序，想要在另一台计算机上执行，必须另编程序，造成了大量的重复工作。但由于使用的是针对特定信号计算机的语言，故而运算效率是所有语言中最高的。机器语言，是第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用助记符代替操作码，用地址符号或标号代替地址码，也就是说用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串，比如用“ADD”代表加法，“MOV”代表数据传递等，这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而，使用汇编语言编写的程序，机器不能直接识别，要由一种程序将汇编语言翻译成机器语言，这种起翻译作用的程序叫汇编程序，它是系统软件中语言处理系统软件。汇编语言编译器把汇编程序翻译成机器语言的过程称为汇编。

虽然汇编语言比机器语言易于读写、调试和修改，但是汇编语言同样十分依赖机器硬件，移植性不好。不过，汇编语言的效率十分高，具有机器语言的全部优点，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精练而且质量高，所以至今仍是一种常用且强有力的软件开发工具。

3. 高级语言

不论是机器语言还是汇编语言，都是面向硬件的具体操作的，语言对机器过分依赖，使用者必须对机器结构及其工作原理非常熟悉，但这对于非计算机专业的人来说，是很难做到的，因此对计算机的推广非常不利。从最初这些与计算机交流的痛苦经历中，人们认识到必须有这样一种新的语言，它不依赖于计算机硬件，能在大部分的机器上通用，比较接近数学语言或人的自然语言，也就是第三代计算机语言——高级语言。

我们知道计算机只能识别机器语言，因此要想让计算机能理解高级语言程序，必须给它配备翻译程序，按照方法不同，翻译程序可分为解释程序和编译程序。

解释程序：所谓解释程序，它将源语言（如 BASIC）书写的源程序作为输入，解释一句后就提交计算机执行一句，并不形成目标程序。就像外语翻译中的“口译”一样，说一句翻译一句，不产生全文的翻译文本。这种工作方式非常适合于小型机的计算问题，但解释程序执行速度很慢，例如，源程序中出现循环则解释程序也重复地解释并提交执行这一组语句，

从而造成很大浪费。

编译程序：编译程序是一类很重要的语言处理程序，它把高级语言（如 Fortran，COBOL，C 等）源程序作为输入，进行翻译转换，产生出机器语言的目标程序，然后再让计算机去执行这个目标程序，得到计算结果。就像外语翻译中的“笔译”一样，要把全文的翻译完成后，形成全文的翻译文本，才可以提交作业。

在实际应用中，对于需要经常使用的有大量计算的大型题目，采用速度较快的编译型的高级语言较好，虽然编译过程本身较为复杂，一旦形成目标文件，以后可多次使用。相反，对于小型题目或计算简单不太费机时的题目，则多选用解释型的会话式高级语言，如 BASIC，这样可以大大缩短编程及调试的时间。

常用的高级语言主要有以下 5 种：

BASIC：初学者语言，特点是简单易学。

FORTRAN：最早的高级语言，通常用于科学计算。

COBOL：商业语言，主要用于商务数据的处理。

Pascal：结构化程序设计语言，主要用于程序设计教学。

C：应用最广泛的程序设计语言，它具有良好的结构，编译后代码执行效率高，不仅具有高级语言的全部优点，还能完成一些底层的操作。

1.2 C 语言及其特点

1.2.1 C 语言的发展简史

C 语言是一种广泛应用于专业程序设计中的高级程序设计语言。它是在 B 语言的基础上发展起来的，其根源可以追溯到 1960 年出现的 ALGOL 60。ALGOL 60 是一种算法语言，它离硬件比较远，不宜用来编写系统程序。1963 年，英国剑桥大学推出了 CPL（Combined Programming Language）语言。CPL 语言比 ALGOL 60 接近硬件，但规模较大，难以实现。1967 年，英国剑桥大学的 M.Richards 对 CPL 语言做了简化，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，设计出了简单而又接近硬件的 B 语言（取 BCPL 的第一个字母），并用 B 语言编写了 UNIX 操作系统，但 B 语言过于简单，功能有限。1972 年至 1973 年间，AT&T 公司 Bell 实验室的丹尼斯·里奇（D.Ritchie）在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母）。C 语言既保持了 BCPL 和 B 语言的优点（精练、接近硬件等），又克服了它们的缺点（过于简单、数据无类型等）。

最初 C 语言只是为了描述和实现 UNIX 操作系统提供一种工作语言而设计的，首先在 DEC 公司的 PDP-11 机上实现。1973 年，K.Thompson 和 D.Ritchie 两人合作用 C 语言改写了 90%以上的 UNIX。经多次改进直到 1975 年，UNIX 第 6 版发布后，C 语言的突出特点才引起人们的普遍关注。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》，并迅速地推动了 UNIX 操作系统在各种机器上的安装（可以不经过较大地改动而方便地从一个平台移植到另一个平台）。后来随着 UNIX 日益广泛的使用，C 语言也迅速得到推广。1978 年出现了 UNIX 第 7 版，以此系统的 C 编译程序为基础，B.Kernighan 和 D.Ritchie

（合称 K&R）合著了影响深远的《The C Programming Language》，它被称为标准 C。1983 年，美国国家标准化协会（ANSI）对标准 C 进行了发展和扩充，制定了新的标准，称为 ANSI C。1987 年，ANSI 又公布了新标准——87 ANSI C。1988 年，K&R 根据 ANSI C 标准重新编写了《The C Programming Language》。1990 年，国际标准化组织 ISO（International Standard Organization）接受 87 ANSI C 为 ISO C 的标准（ISO9899–1990）。目前广泛流行的 C 编译系统大多都是以它为基础的。

C 语言是一种通用的程序设计语言，它糅合了高级语言的一系列特点与汇编语言的高效率特点，并且不专用于某一个特定的应用领域。另外，C 语言限制少，通用性强，这使得它比一些公认的其他语言用起来更方便、效率更高。目前，C 语言已从位于贝尔实验室的发源地传播到世界各地，成为全球程序员的公共语言，并由此诞生了几个新的主流语言 C++，Java 等，它们都建立在 C 语言语法和基本结构的基础上。现在国际上的许多软件都是在 C 语言及其衍生的各种语言的基础上开发出来的。

1.2.2 C 语言的特点

一种语言之所以能存在和发展，并具有较强的生命力，总是有其不同于（或优于）其他语言的特点。C 语言的主要特点如下：

1. 语言简洁、紧凑、使用方便、灵活

ANSI C 语言共有 32 个关键字，5 种基本语句。语句规模小，相对简单，表示方法简洁，高度灵活。

2. 运算符丰富

C 语言的运算符有 30 多种，除此之外，括号、逗号、强制类型转换等都作为运算符处理，从而使 C 的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据类型丰富，具有现代化语言的各种数据结构

C 语言的数据类型不仅包括整型、实型、字符型等基本类型，而且还包括数组、结构体、共用体等构造类型和指针类型。尤其是指针类型的使用，可实现各种复杂数据结构（如链表、栈、树、图等）的运算。

4. 具有结构化的控制语句

C 语言有 3 种基本控制结构（顺序结构、分支结构和循环结构），并且用函数作为程序模块单位以实现程序的模块化。因此 C 语言是结构化的理想语言，符合现代编程风格要求。

5. 语法限制不太严格，程序设计自由度大

C 语言语法限制不太严格，使程序设计比较自由。例如，C 语言对数组下标不做检查，而是由编程人员确保程序的正确性；对变量的类型使用也比较灵活，例如整型与字符型数据可以通用。

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，因此C既具有高级语言的功能，又具有低级语言的许多功能，可用来写系统软件。C语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言。有人把C称为“高级语言中的低级语言”，也有人称它为“中级语言”，意为兼有高级和低级语言的特点。

7. 生成目标代码质量高，程序执行效率高

一般只比汇编程序生成的目标代码效率低10%～20%。

8. 可移植性好

C程序基本上不作太多修改就能用于各种类型的计算机和各种操作系统，原因是C语言中没有依赖于硬件的输入/输出语句，输入/输出操作均是通过调用系统提供的标准库函数来实现的。

1.3 C语言程序的运行环境

在较早期的程序设计中，各个阶段都要用不同的软件来进行处理，如先用字处理软件编辑源程序，然后用连接程序进行函数、模块连接，再用编译程序进行编译。因此，开发者必须在几种软件间来回切换操作。为了简化开发者的操作，当前的编程开发软件将编辑、编译、调试等功能集成在一个桌面环境中，这样就大大方便了用户。如Windows应用程序的开发工具有Visual C++、Visual Basic、Dephi等。

本书所有的程序都在Visual C++ 6.0集成开发环境中编辑、编译、连接和运行。Visual C++提供了一个支持可视化编程的集成开发环境Visual Studio（又名Developer Studio）。Developer Studio是一个通用的应用程序集成开发环境，不仅支持Visual C++，还支持Visual Basic、Visual J++、Visual InterDev等Microsoft系列开发工具。Developer Studio包含文本编辑器、资源管理器、工程编译工具、增量连接器、源代码浏览器、集成调试工具以及一套联机文档。使用Developer Studio可以完成创建、调试、修改应用程序等各种操作。

读者还可以选择Turbo C 2.0等集成开发环境来编辑、编译、连接和运行C程序。

1.4 开发一个简单的C程序

1.4.1 C程序的开发过程

开发一个C程序一般包括以下几个步骤（如图1.1所示）。

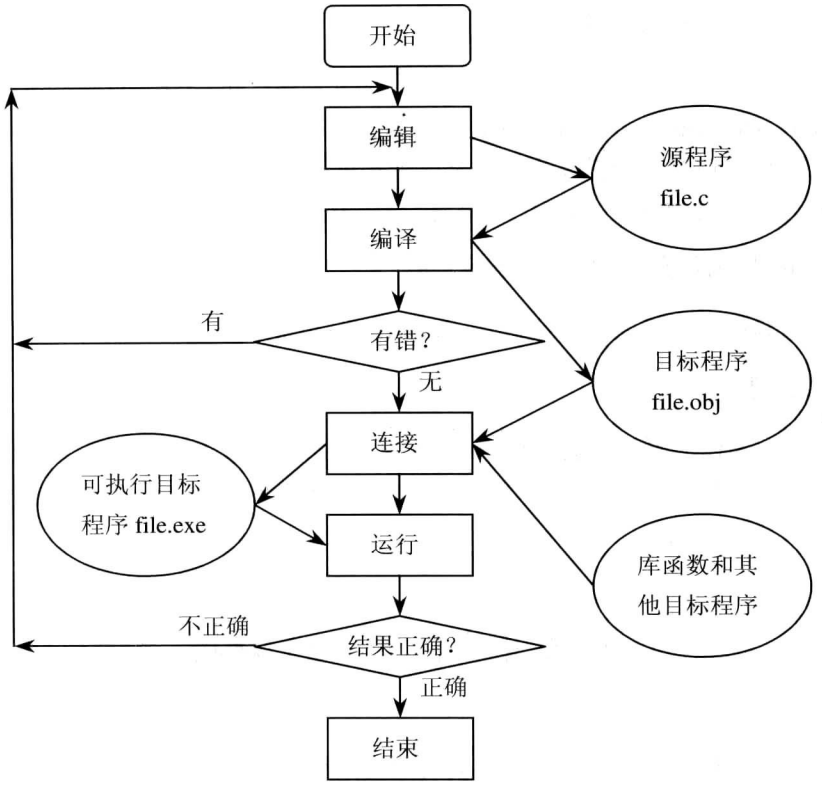

图 1.1　C 程序开发步骤

（1）编辑。程序员使用任意编辑软件（编辑器）将编写好的 C 程序输入计算机，构成源程序文件，存盘时文件名应以.c 为扩展名。

（2）编译。编译时，编译器首先会检查源程序中每一条语句的语法。当发现语法错误时，就在屏幕上显示错误的位置和错误类型的信息。此时，要再次调用编辑器进行查错并修改。然后，再进行编译，直至排除所有语法和语义错误。C 语言程序在编译过程中发现的错误通常可以分成两类：一类是局部语法错误，如拼写错误、少写了必要的符号等；另一类是程序里上下文关系方面的错误，如使用的变量没有预先定义等。编译通过后，编译系统将源程序编译成目标程序。

（3）连接。编译后产生的目标文件是可重定位的程序模块，不能直接运行。连接是将目标程序、库函数或其他目标程序连接组装起来，形成可直接执行的绝对指令代码程序，即生成可执行文件。其主文件名与源文件名相同，扩展名为.exe。

（4）测试与调试。一个程序经过编译、连接产生了可执行程序后，就可以开始测试和调试了。通常可以输入一些实际数据来验证程序执行结果的正确性。如果程序执行中出现问题，或者发现程序的输出结果不正确，就需要设法找到出错的原因，并修改程序、重新编译、重新连接，再测试调试，不断反复，直到确信程序正确无误。

（5）运行。生成可执行文件后，程序就可以在操作系统控制下运行了。

大部分 C 语言都提供一个独立的集成开发环境，可以将上述 5 步连贯在一个程序之中。

本书所涉及的程序全部在 Visual C++ 6.0 集成开发环境中运行。

1.4.2 简单C程序介绍

下面介绍几个简单的C程序，然后从中分析C程序的特性。

【例 1.1】 在屏幕上输出“hello C world!”。

```
main()
{
    printf("hello C world!\n");
}
```

本程序的运行结果如下：

```
hello C world!
```

其中 main 表示“主函数”，每一个 C 程序都必须有一个 main 函数。函数体由花括号“{ }”括起来，其中只有一个输出语句 printf（详见第 3 章）。“\n”是换行符。

【例 1.2】 求两个数的和。

```
main()
{
 int a,b,sum;
 a=123;b=456;
 sum=a+b;                                  /*求 a+b 的和，把值赋予 sum*/
 printf("sum of a plus b is %d\n",sum);
}
```

本程序中，/*…*/表示注释部分，为便于理解。注释只是给人看的，对编译和运行不起作用，注释可以加在程序中任何位置。第三行是变量定义部分，说明 a、b、sum 是整型（int）变量。第四行是两个赋值语句，使 a 和 b 的值分别为 123 和 456。第五行使 sum 的值为 a+b。第六行中的“%d”是输入输出“格式字符串”，用来指定输入输出时的数据类型和格式（详见第 3 章），“%d”表示“十进制整数类型”，在执行输出时，此位置上代以一个十进制整数，因此本程序的运行结果如下：

```
sum of a plus b is 579
```

【例 1.3】 输出两个数中最大的一个。

```
main()                                    /*主函数*/
{
 int a,b,c;                               /*定义变量*/
 printf("input a,b: ");
 scanf("%d,%d",&a,&b);                    /*输入变量 a 和 b 的值*/
 c=max(a,b);                          /*调用 max 函数，将得到的值赋给 c*/
 printf("Max of a and b is %d\n",c);/*输出 c 的值*/
}
```

```
int max(int x,int y)                    /*定义 max 函数，函数值为整型，x,y 为形式参数*/
{
int z;                                  /* max 函数中用到的变量 z，也要加以定义*/
if(x>y) z=x;
else z=y;
return (z);                             /*将 z 的值返回，通过 max 带回调用处*/
}
```

本程序包括两个函数：主函数和被调函数 max，max 函数的作用是将 x 和 y 中较大者的值赋给变量 z，return 语句将 z 的值返回给主调函数 main。返回值是通过函数名 max 带回到 main 函数的调用处。main 函数中的 scanf 函数是“输入函数”的名字（详见第 3 章），此函数的作用是将两个数值分别输入给变量 a 和 b。第六行调用 max 函数，在调用时将实际参数 a 和 b 的值分别传送给 max 函数中的形式参数 x 和 y。经过执行 max 函数得到一个返回值（即 max 函数中变量 z 的值），把这个值赋给变量 c，然后输出 c 的值。程序运行情况如下：

```
8，5↙
Max of a and b is 8
```

本书中用加粗的形式表示输入的部分。

通过上面这些例子，可以看到：

1. C 程序是由函数构成的

一个 C 源程序至少且仅包含一个函数（main 函数），也可以包含一个 main 函数和若干其他函数。C 的函数相当于其他语言中的子程序。可以说 C 是函数式语言，它用函数来实现特定的功能，程序全部工作都是由函数来完成的。函数的位置没有硬性规定，函数之间可以相互嵌套调用，但不能嵌套定义。

2. 一个函数由两部分组成

一个函数由函数的说明部分和函数体组成。

3. 一个 C 程序总是从 main 函数开始执行的

不论 main 函数在整个程序中的位置如何（main 函数可以放在程序最前头，也可以放在程序最后，或在一些函数之前在另一些函数之后），一个 C 程序总是从 main 函数开始执行。

4. C 程序书写格式自由

书写 C 程序时，一行内可以写几个语句，一个语句也可以分几行来写，而且程序中没有行号。

5. 每个语句和数据定义的最后必须有一个分号

分号是 C 语句的必要组成部分。例如：

```
c=a+b;
```

分号不可少。即使是程序中的最后一个语句也应包含分号。