



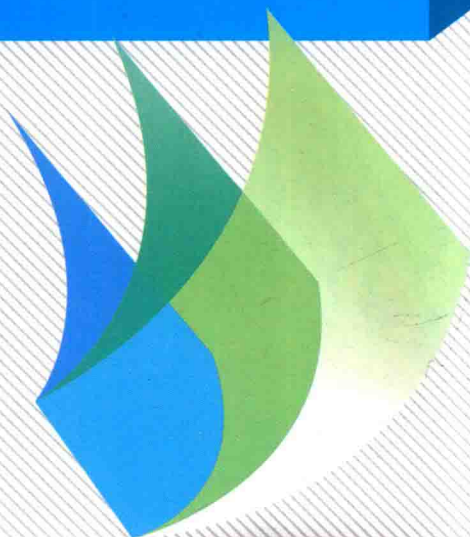
国家精品课程教材
国家级精品资源共享课教材
普通高等教育“十三五”规划教材
研究型教学模式系列教材

C语言程序设计 (第3版)

■ 蒋彦 韩玫瑰 主编 ■ 刘明军 主审



INFORMATION
TECHNOLOGY



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

国家精品课程教材
国家级精品资源共享课教材
普通高等教育“十三五”规划教材
研究型教学模式系列教材

C 语言程序设计

(第3版)

蒋彦 韩玫瑰 主编

张芊茜 黄艺美 崔忠玲 李崇威 编

刘明军 主审



INFORMATION
TECHNOLOGY

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是国家精品课程教材、国家级精品资源共享课教材，以任务驱动的方式，通过实例讲授程序设计的基本概念和基本方法，把重点放在解题思路上，试图贯穿以程序编写带动语法教学的模式，引导读者掌握 C 语言的核心编程方法，提高应用能力。本书共 7 章，主要内容包括：C 语言程序基础、程序基本结构、模块化程序设计、简单构造数据类型、复杂构造数据类型、磁盘数据存储、实用程序设计技巧等。本书配套《C 语言程序设计实验教程（第 3 版）》，并提供课程网站、电子课件、习题答案及程序源代码。

本书可作为高等学校本科生教材，也可作为专科和高职高专教材及计算机等级考试的参考书，还可供相关领域的工程技术人员学习参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

C 语言程序设计/蒋彦，韩玫瑰主编. —3 版. —北京：电子工业出版社，2018.3

ISBN 978-7-121-33770-3

I. ①C… II. ①蒋… ②韩… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字（2018）第 037875 号

策划编辑：王羽佳

责任编辑：王羽佳

印 刷：北京京师印务有限公司

装 订：北京京师印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1 092 1/16 印张：14 字数：455 千字

版 次：2007 年 1 月第 1 版

2018 年 3 月第 3 版

印 次：2018 年 3 月第 1 次印刷

定 价：39.90 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254535 wyj@phei.com.cn。

第3版前言

济南大学开设“C 语言程序设计”课程已有 20 多年的历史，在课程组全体老师的共同努力下，该课程 2005 年被评为山东省精品课程，2009 年被评为国家精品课程，2016 年被评为国家级精品资源共享课。

2007 年我们编写了本课程教材《C 语言程序设计》，2011 年修订了《C 语言程序设计（第 2 版）》。教材出版以后，被多所高等学校选作教材，并被数十所高校选作教学的主要参考书。近年来，陆续收到了各位同仁和广大读者给予的高度评价，以及一些很好的修订建议。七年后，我们根据在教学过程中的实际感受，结合收集到的建议和意见，对第 2 版教材进行了修订，出版了《C 语言程序设计（第 3 版）》和《C 语言程序设计实验教程（第 3 版）》。

第 3 版的主要修订内容体现在以下几个方面：

① 作为《大学计算机》课程的衔接，用 C 语言实现《大学计算机》课程中的算法与程序设计思想，有利于加强学生计算思维能力的锻炼与培养。

② 结合 ACM 竞赛的参赛经验，把 ACM 竞赛模式引入到教学中，建立了适合本课程教学的 OJ（Online Judge，在线判题）系统，把课程中的全部例题、习题、实验改编为 ACM 模式，提供在线练习，系统实时返回运行结果。

③ 指针作为 C 语言的特色，也是该课程的难点之一，大多教材都作为独立的一章进行讲解。我们认为指针也是变量中的一种，只是有其特殊性。因此，将指针的概念和有关内容分布到教材的多个章节中进行讲解，淡化了指针的独立性。希望这样更有利于难点的分解，有助于读者逐步建立起指针的概念，掌握指针的使用方法。

④ 从模块化程序设计的思想引入函数，调整了全书结构，使结构层次更清晰，并以实例的方式说明模块化程序设计的优点。对于文件操作，增加了实用例题，使其更具有指导性。实用程序设计技巧更侧重于实用性和技巧性，并针对 Visual C++ 6.0 环境设计了例题。

⑤ 将程序移植到 Visual C++ 6.0 环境中调试、运行。所有例题均在 Visual C++ 6.0 下调试通过。

本书配套《C 语言程序设计实验教程（第 3 版）》，并提供电子课件、习题参考答案及程序源代码，请登录华信教育资源网 <http://www.hxedu.com.cn> 注册下载。

本书由蒋彦、韩玫瑰统稿，其中第 1~2 章由蒋彦、刘明军修订，第 3 章由张芊茜、李崇威修订，第 4~5 章由崔忠玲、黄艺美修订，第 6~7 章由韩玫瑰修订，全部习题由蒋彦修订。全书由刘明军教授审定。

在本书的修订过程中，得到了全体课程组老师的关心与支持。本书的编写参考了大量近年来出版的相关书籍及技术资料，吸取了许多专家和同仁的宝贵经验。在此一并表示衷心的感谢！

尽管我们做出了很大的努力，修订了第 2 版的诸多不足之处，但由于水平有限，书中难免还有错误或不妥之处。我们恳请各位专家、同行及广大读者批评指正！

作 者

2018 年 2 月

第2版前言

济南大学开设“C 语言程序设计”课程已有十余年的历史，在课程组全体成员的共同努力下，该课程 2005 年被评为山东省精品课程，2009 年被评为国家精品课程。

2007 年我们编写了本课程教材《C 语言程序设计》，在电子工业出版社出版。该教材出版以后，被多所高等学校选作教材，并被数十所高校作为教学的主要参考书使用。在使用过程中，各位同仁和广大读者给予了较高的评价，并提出了很好的建议。四年后，我们根据在教学过程中的实际感受，结合收集到的建议和意见，对第 1 版教材进行了修订，出版了《C 语言程序设计（第 2 版）》和《C 语言程序设计实验教程（第 2 版）》。

第 2 版的主要修订内容体现在以下几个方面：

① 指针作为 C 语言的特色，也是该课程的难点之一，一般的教材都作为独立的一章进行讲解。我们认为指针也是变量中的一种，只是有其特殊性。因此将指针的概念和有关内容分布到教材的多个章节中进行讲解，淡化了指针的独立性。希望这样可以有利于难点的分散，有助于读者逐步建立起指针的概念，掌握指针的使用方法。

② 从模块化程序设计的思想引入函数，调整了全书结构，使结构层次更清晰，并以实例的方式说明模块化程序设计的优点。

③ 对于构造数据类型，除更清楚地介绍基本概念和使用方法外，重点介绍与指针的关系，并结合实例，突出将其作为函数参数的使用方法，与指针内容紧密结合。

④ 对于文件操作，增加了实用例题，使其更具有指导性。实用程序设计技巧更侧重于实用性和技巧性，并针对 Visual C++ 6.0 环境设计了例题。

⑤ 将程序移植到 Visual C++ 6.0 环境中运行、调试。所有例题均在 Visual C++ 6.0 下调试通过。

本书配套《C 语言程序设计实验教程（第 2 版）》，并提供电子课件、习题参考答案及程序源代码，请登录华信教育资源网 <http://www.hxedu.com.cn> 注册下载，或登录国家精品课程网站 <http://c.ujn.edu.cn/>。

本书由刘明军统稿，其中第 1~2 章由刘明军修订，第 3 章由李崇威修订，第 4~5 章由黄艺美修订，第 6~7 章由韩玫瑰修订，部分章节及全部习题由蒋彦修订。

在本书的修订过程中，得到了全体课程组教师的关心与支持。本书的编写参考了大量近年来出版的相关书籍及技术资料，吸取了许多专家和同仁的宝贵经验。在此一并表示衷心的感谢！

尽管我们做出了很大的努力，修订了第 1 版的诸多不足之处，但由于水平有限，书中难免还有错误或不妥之处。我们恳请各位专家、同行及广大读者批评指正！

作者

2011 年 1 月

前 言

C 语言是一种被广泛学习、普遍使用的计算机程序设计语言。它的高级语言形式、低级语言功能的特点具有特殊的魅力，因而被大多数高等学校采用作为典型的计算机语言教学课程，也被选为计算机等级考试、全国计算机应用证书考试等多种计算机技能考试的考试内容。此外，C 语言作为一门实用且功能强大的程序设计语言，被程序设计人员广泛使用。因此，C 语言是一门十分重要的程序设计语言。

学习程序设计语言的目的在于使用该语言编写程序解决实际工作中的问题，而不仅仅是掌握该语言的语法。读者在 C 语言程序设计的学习中，很容易陷入语法的泥潭中，一开始就学习大量的语法知识，难以记忆，更难以理解，难免产生为难情绪，感到 C 语言难学，更不知如何使用。作者在建设“省级精品课程”的同时，对课程的教学理念和教学方法进行了认真思考，集聚多年的教学经验，从而完成了本书——一本真正注重培养读者程序设计能力的教材。

本教材具有以下特色：

① 进行了基于任务驱动的实例教学法的尝试。以任务驱动的方式，通过实例讲授程序设计的基本概念、基本方法，把重点放在解题思路上。

② 从开始学习就使读者将注意力集中在所解决的问题领域，从具体实例理解 C 语言的开发特点和程序总体框架，通过实例本身学习某一类问题的解决方法和计算设计，贯穿以程序编写带动语法教学的模式。

③ 在 C 语言的环境下，针对实际问题进行分析，构建数学模型，设计算法，最后编程实现。

④ 将原来的被动填鸭式的灌输语言知识，变为自主的学习和探索，读者不再陷入语言的语法规则，而重在程序设计方法的学习和探究，通过编程掌握语言。

⑤ 在学习的不同阶段设计不同的针对性的实际，从而得到较好的教学效果。例如，开始阶段设计的实例是将学生的注意力吸引在 C 语言的总体功能和程序的总体框架上；在学习中间阶段设计针对某些数据类型或应用特点的实例、针对模块分解和组合的实例、针对算法分析与设计的实例等；在学习的后期进行综合课程设计，将所学知识融会贯通。

⑥ 引导读者掌握 C 语言的核心编程方法，提高应用能力。引导学生在解题编程的实践中探索其中的规律，将感性认识升华到理性高度。

全书分为理论部分和实验部分。理论部分共分 7 章。第 1 章介绍 C 语言的基本结构、语法成分、调试方法，简单 C 语言程序的设计、输入/输出语句等；第 2 章结合实际问题的 C 语言程序的基本结构；第 3 章介绍如何将复杂问题简单化处理的编程方法；第 4、5 章介绍如何编写具有构造数据类型的程序；第 6 章介绍磁盘数据存储程序的设计方法；第 7 章介绍实用程序设计的一般方法。

通过学习本书，你可以：

- 以任务驱动的方式了解 C 语言程序设计的基础知识

- 掌握 C 语言的核心编程方法
- 以程序编写带动语法的学习
- 建立程序设计的思想
- 通过上机实验提高程序设计能力
- 小试身手——利用 C 语言进行程序设计

本书的编写工作由刘明军主持。第 1、2 章由刘明军编写，第 3 章由王卫峰编写，第 4、5 章由黄艺美编写，第 6、7 章由韩玫瑰编写，实验部分由蒋彦和韩玫瑰老师编写，潘玉奇老师参加了部分内容及部分习题的编写。全书由刘明军统稿。中国石油大学（华东）的魏东平教授在百忙之中审阅了全书，济南大学的董吉文教授对本书的编写提出了宝贵意见。本书的编写参考了近年来出版的大量书籍及相关技术资料，吸取了许多同仁和专家的宝贵经验，在此一并表示衷心的感谢！

在琳琅满目的书海中，编写一本有特色并能使读者感兴趣的教材绝非易事。尽管我们付出了很大的努力，但由于水平有限，书中难免出现错误或不妥之处，我们诚恳地欢迎读者和同行批评指正。

作 者

2007 年 1 月

目 录

第 1 章 C 语言程序基础	(1)	2.1.6 switch 语句	(39)
1.1 C 语言程序的基本结构	(1)	2.2 关系运算和逻辑运算	(40)
1.1.1 认识 C 语言程序	(1)	2.2.1 关系运算符和关系表	
1.1.2 基本结构	(3)	达式	(41)
1.1.3 程序举例	(4)	2.2.2 逻辑运算符和逻辑表	
1.2 C 语言基本语法成分	(6)	达式	(41)
1.3 C 语言数据类型	(8)	2.3 循环结构	(42)
1.3.1 基本数据类型	(9)	2.3.1 概述	(42)
1.3.2 指针类型	(11)	2.3.2 当型循环 while	(43)
1.3.3 构造数据类型	(12)	2.3.3 直到型循环 do-while	(45)
1.4 C 语言的表达式和语句	(12)	2.3.4 当型循环 for	(46)
1.5 C 语言程序运行过程	(13)	2.3.5 几种循环的比较	(48)
1.5.1 源程序、目标程序和可执		2.3.6 循环嵌套	(49)
行程序的概念	(13)	2.4 break 语句和 continue 语句	(50)
1.5.2 C 语言程序的开发步骤	(14)	2.4.1 break 语句	(50)
1.6 编写简单的 C 语言程序	(15)	2.4.2 continue 语句	(51)
1.7 数据的输入与输出	(16)	2.5 goto 语句	(52)
1.7.1 printf 函数	(17)	2.6 指针程序设计	(53)
1.7.2 scanf 函数	(18)	2.6.1 指针	(53)
1.8 C 语言中的宏定义	(20)	2.6.2 指针变量的使用	(54)
1.8.1 不带参数的宏定义	(20)	2.7 典型例题	(56)
1.8.2 带参数的宏定义	(21)	习题	(61)
1.9 算法	(23)	第 3 章 模块化程序设计	(67)
1.9.1 算法的概念及特性	(23)	3.1 模块化程序设计思想	(67)
1.9.2 算法的表示方法	(24)	3.2 函数定义	(69)
1.10 C 语言的产生、发展及特点	(26)	3.3 函数调用	(70)
1.10.1 C 语言的产生及发展	(26)	3.3.1 函数调用的形式	(70)
1.10.2 C 语言的特点	(27)	3.3.2 函数间的参数传递	(71)
习题	(28)	3.3.3 函数的返回值	(73)
第 2 章 程序基本结构	(33)	3.4 函数的原型与声明	(73)
2.1 分支结构	(33)	3.5 函数的嵌套与递归	(75)
2.1.1 单分支结构	(33)	3.5.1 函数的嵌套调用	(75)
2.1.2 双分支结构	(34)	3.5.2 函数的递归调用	(76)
2.1.3 多分支结构	(35)	3.6 库函数	(77)
2.1.4 if 语句的嵌套	(36)	3.7 变量的作用域与存储类型	(78)
2.1.5 条件运算符	(38)	3.7.1 变量的作用域	(78)

3.7.2 变量的存储类型.....	(81)	5.2 共用体.....	(149)
3.8 指针与函数.....	(84)	5.2.1 共用体的定义和引用.....	(149)
3.8.1 指针作为函数参数.....	(84)	5.2.2 共用体类型的特点.....	(150)
3.8.2 返回指针值的函数.....	(87)	5.2.3 共用体应用举例.....	(151)
3.8.3 指向函数的指针变量.....	(88)	5.3 枚举类型.....	(152)
3.9 典型例题.....	(89)	5.3.1 枚举类型的定义和引用.....	(152)
习题.....	(93)	5.3.2 枚举类型应用举例.....	(153)
第4章 简单构造数据类型.....	(98)	*5.4 链表.....	(154)
4.1 一维数组.....	(98)	5.4.1 概述.....	(154)
4.1.1 一维数组的引出.....	(98)	5.4.2 简单链表.....	(155)
4.1.2 一维数组的定义和引用.....	(99)	5.4.3 动态链表.....	(155)
4.1.3 一维数组程序举例.....	(100)	5.4.4 链表的实现及应用.....	(156)
4.2 二维数组.....	(102)	习题.....	(159)
4.2.1 二维数组的引出.....	(102)	第6章 磁盘数据存储.....	(163)
4.2.2 二维数组的定义和引用.....	(103)	6.1 将数据写入文件.....	(163)
4.2.3 二维数组程序举例.....	(104)	6.1.1 打开文件函数.....	(164)
4.3 字符数组与字符串.....	(106)	6.1.2 关闭文件函数.....	(164)
4.3.1 字符数组的引出.....	(106)	6.2 文件读写分类函数.....	(165)
4.3.2 字符数组的定义和引用.....	(107)	6.2.1 单字符写入函数.....	(166)
4.3.3 字符串的使用.....	(108)	6.2.2 单字符读取函数.....	(166)
4.3.4 字符数组程序举例.....	(112)	6.2.3 字符串读取函数.....	(167)
4.4 数组与指针.....	(114)	6.2.4 字符串写入函数.....	(168)
4.4.1 一维数组与指针.....	(114)	6.2.5 格式化读写函数.....	(168)
4.4.2 多维数组与指针.....	(116)	6.2.6 数据块读写函数.....	(170)
4.4.3 数组作为函数参数.....	(117)	6.3 文件定位函数.....	(172)
4.5 字符串与指针.....	(121)	6.3.1 位置指针复位函数.....	(172)
4.5.1 用字符指针访问字符串.....	(121)	6.3.2 位置指针的随机移动函数.....	(174)
4.5.2 字符指针和字符数组的区别.....	(122)	6.3.3 文件指针当前位置函数.....	(175)
4.5.3 字符串作为函数参数.....	(124)	6.4 其他文件函数.....	(176)
4.6 典型例题.....	(125)	6.4.1 文件结束检测函数.....	(176)
习题.....	(132)	6.4.2 出错检测函数.....	(177)
第5章 复杂构造数据类型.....	(139)	习题.....	(177)
5.1 结构体.....	(139)	第7章 实用程序设计技巧.....	(181)
5.1.1 结构体的引出及使用.....	(139)	7.1 程序的模块化结构.....	(181)
5.1.2 结构体数组.....	(143)	7.1.1 软件工程的思想.....	(181)
5.1.3 结构体程序举例.....	(145)	7.1.2 模块设计.....	(181)
5.1.4 结构体与指针.....	(146)	7.1.3 模块化的优点.....	(182)
		7.2 模块的组装.....	(183)

7.2.1 文件包含与头文件的使用	(183)	附录	(203)
7.2.2 模块间的连接	(185)	附录 A 常用 C 语言库函数	(203)
7.2.3 标识符的一致性	(188)	A.1 数学函数	(203)
7.2.4 条件编译	(188)	A.2 输入/输出函数	(203)
7.3 模块设计风格简述	(190)	A.3 字符函数	(206)
7.3.1 数据风格	(190)	A.4 字符串函数	(207)
7.3.2 标识符风格	(190)	A.5 动态存储分配函数	(207)
7.3.3 算法风格	(191)	A.6 时间函数	(208)
7.3.4 输入/输出风格	(191)	A.7 其他函数	(209)
7.3.5 书写风格	(192)	附录 B ASCII 码表	(210)
7.4 应用程序设计实例	(193)	附录 C C 语言运算符的优先级与 结合性	(212)
习题	(200)	参考文献	(214)

第1章 C语言程序基础

计算机之所以能够自动、高速地进行大量的计算、处理各种信息，源于事先存储的程序。计算机的所有操作都是在程序的控制之下完成的，计算机离不开程序。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的集合。

C语言是计算机最常用的程序设计语言，是国际上最流行的、应用面最广的高级程序设计语言。C语言也是一种结构化程序设计语言，用它编写的程序层次清晰，便于按模块化方式组织，易于调试和维护。40多年来，C语言经历了不断的发展和完善，已经成为国内外公认的一种优秀的程序设计语言，有着其他语言不可比拟的优点。

本书以C语言程序设计为主线，介绍程序设计的基本概念和基本方法。

1.1 C语言程序的基本结构

1.1.1 认识C语言程序

为说明C语言程序的结构，首先来看下面简单的C语言程序。

【例1.1】 在屏幕上输出一行信息：This is a C program.

```
#include <stdio.h>          /* 预处理命令 */
int main()                  /* 定义主函数 */
{
    printf("This is a C program.\n"); /* 调用库函数，输出信息 */
    return 0;               /* 返回0 */
}
```

【例1.2】 计算两数之和，并输出结果。

```
#include <stdio.h>          /* 预处理命令 */
int main()                  /* 定义主函数，计算两数之和 */
{
    int a,b,sum;            /* 定义三个整型变量 */
    a=123; b=456;          /* 为变量a, b赋值 */
    sum=a+b;                /* 让sum等于a+b的值 */
    printf("sum=%d\n",sum); /* 调用库函数，输出结果 */
    return 0;              /* 返回0 */
}
```

现在对这两个C语言程序进行解析。

① `#include <stdio.h>`是预处理命令，表示文件包含，其功能是将头文件 `stdio.h` 的内容包含到用户源程序中。文件 `stdio.h` 中声明了程序所需要的输入和输出操作的有关信息，有了该预处理命令后，程序中就可以使用包含文件的内容了。

② `main` 表示主函数，`main` 是函数名。每个C语言程序必须有 `main` 函数。函数名后的一对圆括号不能省略。一个C语言程序可以包含若干个函数，但只能有一个主函数。`main` 前面的 `int` 是一种数据类型，说明函数执行完毕后向系统返回一个整数，具体返回的数值即 `return` 语句后的 `0`，返回 `0` 表示程序正常结束。

③ 用 `{ }` 括起来的是 `main` 的函数体。`main` 函数中的所有操作（语句）都在这一对 `{ }` 之间。也就

是说 main 函数的所有操作都在 main 函数体中。

④ printf 是 C 语言的标准输出函数，其含义是将双引号内的内容输出到显示器屏幕上。例 1.1 中的 printf 用于将字符串 “This is a C program.\n” 输出，即在屏幕上显示：This is a C program。后光标跳到下一行行首。“\n” 是换行符，使光标跳到下一行行首（第一列）。例 1.2 中的 %d 为格式控制符，表示在此位置将输出一个十进制整数，该整数由逗号后面的变量 sum 提供。程序运行后，将在屏幕上输出：sum=579。

⑤ 分号 “;” 是 C 语句结束符，表示该语句结束。

⑥ “/* */” 括起来的部分是一段注释，注释只是为了改善程序的可读性，在编译、运行时不起作用（事实上编译时会跳过注释，目标代码中不会包含注释）。注释可以放在程序的任何位置，并允许占用多行，只是需要注意 “/*” 与 “*/” 匹配且不能嵌套注释。在 Visual C++ 集成开发环境下，注释可写在 “//” 后，但注释内容仅限 “//” 所在行。

⑦ “int a,b,sum;” 是变量声明，声明了 3 个具有整数类型的变量 a、b、sum。C 语言的变量必须先声明后使用。

⑧ “a=123; b=456;” 是两条赋值语句，将 123 赋给变量 a，456 赋给变量 b。执行上述两条语句后，a、b 两个变量的值分别为 123 和 456。

⑨ “sum=a+b;” 将 a、b 两个变量的内容相加，然后将结果赋值给整型变量 sum。此时 sum 的内容为 579。

【例 1.3】 输入两个整数，计算并输出两者中较大的数。

```
#include <stdio.h>
int max(int x,int y)           // 定义 max 函数，用于计算两数中较大的数
{                               // max 函数体开始
    int z;                     // 声明部分，定义变量
    if (x>y) z=x;               // 如果 x>y，则将 x 的值赋给 z
    else z=y;                  // 否则，则将 y 的值赋给 z
    return z;                  // 将 z 值返回，通过 max 带回调用处
}                               // max 函数体结束
int main()                     // 主函数
{                               // main 函数体开始
    int a,b,c;                 // 声明部分，定义变量
    scanf("%d%d",&a,&b);        // 调用库函数，输入变量 a、b 的值
    c=max(a,b);                // 调用 max 函数，将调用结果赋给 c
    printf("max=%d\n",c);      // 调用库函数，输出 c 的值
    return 0;                  // 返回 0
}                               // main 函数体结束
```

说明：

① 本程序包括两个函数。其中，主函数 main 仍然是整个程序执行的起点，函数 max 的功能是计算两数中较大的数。

② scanf 是 C 语言的标准输入函数，用于从键盘输入若干数据给指定变量。%d 表示输入十进制整数。主函数 main 调用 scanf 函数从键盘获得两个整数，存入 a、b 两个变量中，然后调用函数 max 获得两个数中较大的值，并赋给变量 c，最后输出变量 c 的值（结果）。

③ int max(int x,int y) 是函数 max 的函数首部，表明此函数从调用它的函数（此处为 main 函数）获得两个整数，返回一个整数。

④ 用 { } 括起来的部分是 max 函数的函数体。max 的函数体是函数 max 的具体实现，它从参数表获得数据，处理后得到结果 z，然后将 z 返回调用函数 main。

⑤ 本例表明，除了可以调用库函数外，还可以调用用户自己定义、编写的函数。

1.1.2 基本结构

综合上述3个例子，下面对C语言程序的基本组成和形式（程序结构）加以说明。

C语言程序的结构如下：

```
预处理命令
int main()           // 主函数
{                   // 函数体开始
    声明部分
    执行部分
    return 0;
}                   // 函数体结束
其他函数           // 自定义函数
{
    声明部分
    执行部分
}
```

下面对C语言程序的结构进行简单说明。

1. 函数是C语言程序的基本单位

一个C语言源程序必须包含一个main函数，还可以包含一个或多个其他函数。函数是C语言程序的基本单位。C语言是函数式的语言，程序的全部工作都是由各个函数完成的。编写C语言程序就是编写一个个函数。

程序中的函数有些是C语言的标准库提供的，称为标准函数（或库函数），如printf函数和scanf函数。编写程序时，如果有标准函数，就使用标准函数；如果没有标准函数，则需要用户自己编写函数。用户编写的函数可以写在main函数的前面或后面，可由main函数调用，也可以被其他自定义函数调用。通常，每个函数都完成一项独立的功能。

C语言函数库非常丰富，ANSI C提供100多个库函数，Turbo C提供300多个库函数，而Visual C++提供的库函数更多。

不同的编译系统除了提供函数库中的标准函数外，还按照硬件的情况提供一些专门的函数，因此不同编译系统提供的函数数量和功能会有一定差异。

2. main函数

主函数是每个程序执行的起点。每个C语言程序都是从main函数开始执行的，而不论main函数在程序中的位置。可以将main函数放在整个程序的最前面，也可以放在整个程序的最后，或者放在一些函数之后另一些函数之前。但一个程序只能有一个main函数。

3. 函数的结构

无论是main函数还是自定义函数，每个函数都由函数首部和函数体两部分组成。

(1) 函数首部

即一个函数的第一行，格式如下：

返回值类型 函数名([函数参数类型1 函数参数名1], ..., [函数参数类型n 函数参数名n])

例如：

```
int max(int x,int y)
```

注意：一个函数可以没有参数，但是后面的一对括号不能省略，这是格式的规定，如main函数。

(2) 函数体

函数首部下面用一对{ }括起来的部分。如果函数体内有多对{ }，则最外层是函数体的范围。函

数体一般包括声明部分和执行部分。

- 声明部分：定义本函数所使用的变量，并为每个变量分配相应大小的内存单元。变量名是内存单元的符号地址。
- 执行部分：由若干条语句组成的命令序列（可以在其中调用其他函数）。

4. 书写风格

C 语言程序书写格式自由，一行可以写多个语句，一个语句也可以写在多行上。每条语句的最后必须在末尾用分号“;”表示语句的结束。

5. 输入/输出

C 语言本身不提供输入/输出语句，输入/输出操作是通过调用库函数（如 `scanf` 和 `printf` 等）完成的。

由于输入/输出操作涉及具体的计算机硬件，因此把输入/输出操作放在函数中处理可以简化 C 语言和 C 语言的编译系统，便于 C 语言在各种计算机上实现。

6. 注释

注释部分要放在符号“/*”和“*/”之间或者“//”之后并与其同行。为了便于阅读，程序中应适当地加入注释。注释只起帮助阅读和理解程序的作用，不参加程序的编译，也不影响程序的运行。使用注释是编程人员的良好习惯。

实践中，编写的程序往往需要不断地修改和完善，事实上没有一个应用程序是不需要修改和完善的。很多人会发现自己编写的程序在经历了一些时间以后，如果缺乏必要的文档和必要的注释，最后连自己都很难再读懂，需要花费大量时间重新思考和理解原来的程序，因而浪费了大量的时间。如果一开始编程就对程序进行注释，虽然编写程序时麻烦一些，但日后可以节省大量的时间。

7. 预处理命令

预处理命令能够改进程序的设计环境，提高编程效率。C 语言的预处理功能主要包括：宏定义、文件包含和条件编译，分别用宏定义命令（`#define`）、文件包含命令（`#include`）和条件编译命令（`#ifdef...#else...#endif`）实现，为了与一般语句区别，这些命令以“#”开头。

1.1.3 程序举例

【例 1.4】试编写程序，计算圆的周长和面积。

分析：计算圆的周长和面积要用到圆周率 π ，它的值为 3.14，但在 C 语言程序中不能直接使用希腊字母 π ，可以直接使用 3.14，或定义一个常量或变量来表示 π ，本程序中定义了一个变量 `PI`，令 `PI=3.14`。

已知圆的半径为 `r`，则周长 `l=2*PI*r`，面积 `s=PI*r*r`。因为 `PI` 包含小数，所以变量应定义为可以存储小数且精度较高的双精度型变量 `double`。

参考程序如下：

```
#include <stdio.h>
int main()
{
    double PI=3.14,r,l,s;           // 声明 double 型变量
    printf("Input r:\n");           // 输入提示
    scanf("%lf",&r);                 // 输入半径 r
    l=2*PI*r;                        // 计算周长，注意与数学表达式的区别
    s=PI*r*r;                        // 计算面积
    printf("%.2f,%.2f\n",l,s);       // 以 2 位小数形式输出
}
```



```

    return 0;                                // 返回 0
}
Input r:
1.5
9.42, 7.07

```

【例 1.5】 已知 $a=5$, $b=10$, 试交换 a 、 b 的值。

分析: 交换两个变量的值, 就像一个瓶装酱油, 一个瓶装醋, 要交换需要借助第三个空瓶一样, 交换两个数也需要借助第三个变量。

参考程序如下:

```

#include <stdio.h>
int main()
{
    int a=5,b=10,temp;           // 定义变量 a、b 和 temp, 并对 a、b 赋初值
    temp=a;                     // 将变量 a 的值赋给 temp 保存
    a=b;                         // b 的值赋给 a
    b=temp;                     // temp 的值赋给 b
    printf("a=%d,b=%d\n",a,b);  // 输出 a、b 的值
    return 0;
}

```

运行结果:

```
a=10,b=5
```

【例 1.6】 试计算圆柱体的表面积。

分析: 为了计算圆柱体的表面积, 可以利用公式 $2\pi r$ (r 为半径) 得到底面周长 d , 利用公式 $d \times h$ 计算侧面积 s_1 , 利用公式 πr^2 计算底面积 s_2 , 则表面积 $s=s_1+2 \times s_2$ 。因此可以按照以下步骤计算:

- ① 设圆柱体的高为 h , 半径为 r , 表面积为 s ;
- ② 输入 r 、 h 的值;
- ③ 计算底面周长 $d=2\pi r$;
- ④ 计算侧面积 $s_1=dh$;
- ⑤ 计算底面积 $s_2=\pi r^2$;
- ⑥ 计算表面积: $s=s_1+2 \times s_2$;
- ⑦ 输出 s 的值。

这是用自然语言描述的计算圆柱体表面积的算法。在计算过程中, 有些步骤可以合并。现将其转换为 C 语言程序。注意变量 r 、 h 和 s 的数据类型。

参考程序如下:

```

#include <stdio.h>                // 预处理文件包含命令
#define PI 3.1415926             // 预处理, 定义符号 PI 代表常量 3.1415926
int main()                       // main 函数
{
    double r,h,s;                // 声明部分: 定义 3 个变量 r,h,s, 都为实数类型
    printf("Input r,h:\n");      // 执行部分: 输出提示信息
    scanf("%lf%lf",&r,&h);        // 执行部分: 输入 r 和 h 的值
    s=2*PI*r*h+2*PI*r*r;         // 执行部分: 计算表面积
    printf("Total area is %.2f\n",s); // 执行部分: 输出表面积
    return 0;
}

```

说明:

① “`#define PI 3.1415926`” 是宏定义命令, PI 称为符号常量, 即用一个标识符代表一个常量。此处表示用标识符 PI 代表常量 3.1415926。

符号常量的定义形式为:

#define 标识符 常量

② double 表示实型，即定义 r、h 和 s 为实型变量。double 型数据在输入时需要用 %lf 格式控制符，输出时可用 %f 或 %lf 格式控制符，“.2”表示只输出 2 位小数。

1.2 C 语言基本语法成分

1. C 语言字符集

字符是 C 语言最基本的元素，C 语言字符集由字母、数字、空白、标点符号和特殊字符组成（在字符串常量和注释中还可以使用汉字和其他图形符号）。由字符集中的字符可以构成 C 语言进一步的语法成分，如标识符、关键字、运算符等。

C 语言程序是用下列字符所组成的字符集写成的：

① 字母：A~Z，a~z

② 数字：0~9

③ 标点符号、特殊字符：

! # % ^ & + - * / = ~ < > \
 | . , ; : ? ' " () [] { }

④ 空白符：空格、制表符（Tab 跳格键）、换行符（空行）的总称。空白符除了在字符和字符串中有意义外，编译系统忽略其他位置的空白符。空白符在程序中只是起到间隔的作用。在程序的恰当位置使用空白符将使程序更加清晰，增强程序的可读性。

2. 标识符

标识符是用来标识变量名、符号常量名、函数名、数组名、类型名等实体（程序对象）的有效字符序列。标识符由用户定义。

（1）C 语言标识符定义规则

① 标识符只能由字母、数字和下画线 3 种字符组成，且第一个字符必须为字母或下画线。

例如，合法的标识符：a，i，sum，average，_total，Class，day，student，p405；不合法的标识符：5a，M.D.John，\$123，3D64，a-b。

② 大小写敏感，即 C 语言程序认为大小写字母是不同的字符。例如，sum 不同于 Sum，BOOK 不同于 book。一般情况下习惯将变量名小写，常量名大写，但不绝对。

③ ANSI C 没有限制标识符长度，但各个编译系统都有自己的规定和限制（TC 允许用 32 个字符，MS C 只能用 8 个字符）。

例如，student_name 和 student_number，在 MS C 中，系统认为这两个标识符是相同的。

④ 标识符不能与“关键字”同名，也不能与系统预先定义的“标准标识符”同名，如 main、printf 等。

（2）定义标识符需遵循的原则

① 尽量不要用下画线开头。因为系统内部使用了一些下画线开头的标识符（如 _fd、_cs、_ss），要避免与系统定义的标识符冲突。

② 尽量做到“见名知义”，如 sum，area，score，day，name，age 等。

③ 在容易出现混淆的地方应尽量避免使用容易认错的字符。例如，数字 1、字母 l、字母 I，数字 0、字母 o、O，数字 2、字母 Z、字母 z。

3. 关键字

关键字是 C 语言预先定义的、具有特定意义的标识符，也称为保留字。C 语言包括 32 个关键字：

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

C 语言的关键字都是小写的。不能重新定义关键字，也不能把关键字定义为一般标识符。

4. 运算符

运算符是用于描述某种运算功能的符号，如+、-、*、/、%等，可以由一个或多个字符组成。由运算符将常量、变量和函数调用连接起来的式子称为表达式。

根据参与运算的操作数个数，运算符分为：单目（一元）运算符、双目（二元）运算符和三目（三元）运算符。

C 语言运算符分为以下几类：

- 算术运算符：+ - * / %
- 自增、自减运算符：++ --
- 关系运算符：< <= > >= == !=
- 逻辑运算符：! && ||
- 位运算符：<< >> ~ | ^ &
- 赋值运算符：=及扩展赋值运算符
- 条件运算符：?:
- 逗号运算符：,
- 指针和地址运算符：* &
- 求字节运算符：sizeof
- 成员运算符：. ->
- 下标运算符：[]
- 强制类型转换运算符：(类型)

部分运算符功能简介如下。

(1) 算术运算符

C 语言基本的算术运算符共有 5 种：

+（加）、-（减）、*（乘）、/（除）、%（取余，模运算）

它们都是双目运算符，即运算符要求有两个操作数。

- 取余运算“%”左右的两数必须为整型数据，如 7%4 值为 3。
- 两个整数相除，结果为整数，舍去小数部分，如 5/3=1。但是，如果除数或被除数中有一个为负值，则舍入的方向是不固定的。如-5/3 在有的机器上得到的结果是-1，有的机器则是-2。多数机器采取“向零取整”的方法，即取整后向零靠拢（即向实数轴的原点靠拢）。
- 字符型数据可以和数值型数据混合运算。因为字符型数据在计算机内部是用一个字节的整型数表示的，如'A'+32。
- 运算符是有优先级高低之分的，遵循的原则是“先乘除，后加减”。“*、/、%”为同一级别，“+、-”为同一级别，若一个运算对象两侧的运算符优先级别相同时，则按规定的“自左至