

高等学校培养应用型人才教材——计算机系列



实用 C 语言简明教程

顾小晶 主编



中国电力出版社

www.infopower.com.cn

高等学校培养应用型人才教材——计算机系列

实用C语言简明教程

顾小晶 主编

曾海 滕刚 孙浩 刘丽 参编

中国电力出版社

内 容 提 要

本书为高等学校培养应用型人才教材之一，由具有丰富教学经验的一线教师编写。

本书编者根据应用型高校学生的特点，结合多年讲授 C 语言的教学经验，力求做到：注重基础、强调方法、突出实用和强化实践。全书在全面阐述基础语法、语言特点的基础上，强调程序设计的方法和程序设计的风格。全书内容在满足教学大纲和计算机等级考试要求的同时，对于实际应用中 C 语言的不易掌握之处和灵活性加以重点说明。本书重视实践环节，除了每章提供上机实习题外，有专门章节进行上机实验和课程设计指导以及程序调试方法的介绍。同时，在全书的体系结构上也作了精心编排：重点突出、难点分散、深入浅出、循序渐进。

本书是高等院校及相关专业的教材，也可作为程序设计人员学习的参考教材。

图书在版编目 (CIP) 数据

实用 C 语言简明教程 / 顾小晶主编. —北京：中国电力出版社，2003

高等学校培养应用型人才教材——计算机系列

ISBN 7-5083-1524-3

I. 实... II. 顾... 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 (2003) 第 067639 号

中国电力出版社出版、发行

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

汇鑫印务有限公司印刷

各地新华书店经售

*

2003 年 9 月第一版 2003 年 9 月北京第一次印刷

787 毫米×1092 毫米 16 开本 17.5 印张 418 千字

定价 24.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题，我社发行部负责退换)

高等学校培养应用型人才教材——计算机系列

编 委 会

主任委员:

宗 健 常明华

副主任委员:

顾元刚 陈 雁 杨翠南 林全新 华容茂 曹泰斌

魏国英 邵晓根 庄燕滨 邓 凯 吴国经 常晋义

许秀林 谢志荣 张家超 陶 洪 龚兰芳 刘广峰

丁 雁 方 岩 王一曙

委 员: (以姓氏笔画为序)

丁志云 及秀琴 石振国 李 翊 吕 勇 朱宇光

任中林 刘红玲 刘 江 刘胤杰 许卫林 杨劲松

杨家树 杨伟国 郑成增 张春龙 闵 敏 易顺明

周维武 周 巍 胡顺增 袁太生 高佳琴 唐学忠

徐煜明 曹中心 曾 海 颜友钧

序 言

进入 21 世纪,世界高等教育已从精英教育走向了大众教育。我国也适应这一潮流,将高等教育逐步推向大众化。培养应用型人才已成为国家培养国际人才的重要组成部分,且得到了社会各界的广泛支持。于是一大批有规模、有实力、规范化、以培养应用型人才为己任的高等学校得到了长足发展。这类高校办学的一个显著的特点是按照新时代需求和当地的需要来培养学生,他们重视产学研相结合,并紧密地结合当地经济状况,把为当地培养应用型人才作为学校办学的主攻方向。

这类学校的教学特点是:在教授“理论与技术”时,更注重技术方法的教学。在教授“理论与实践”时,更注重理论指导下的可操作性,更注意实际问题的解决。因此,这些学生善于解决生产中的实际问题,受到地方企事业单位的普遍欢迎。

为满足这类高校的教学要求,达到培养应用型人才的目的,根据教育部有关重点建设项目的要求和相关教学大纲,我们组织了多年在这类高校中从教,并具有丰富工程经验的资深教授、高级工程师、教师来编写这套教材。

在这套教材的编写中,我们提倡“实用、适用、先进”的编写原则和“通俗、精练、可操作”的编写风格,以解决多年来在教材中存在的过深、过高且偏离实际的问题。

实用——本套教材重点讲述本行业中最广泛应用的知识、方法和技能。使学生学习后能胜任岗位工作,切实符合当地经济建设的需要和社会需要。

适用——本套教材是以工程技术为主的教材,所以它适用于培养应用型人才的所有高校(包括本科、专科、技术学院、高职等),既符合此类学生的培养目标,又便于教师因材施教。

先进——本套教材所选的内容是当今的新技术、新方法。使学生在掌握经典的技术和方法之后,可用教材中的新技术、新方法去解决工程中的技术难题,为学生毕业后直接进入生产第一线打下坚实的基础。

通俗——本套教材语言流畅、深入浅出、容易读懂。尽量避开艰深的理论和长篇的数学推导,尽量以实例来说明问题,在应用实例中掌握理论,使学生轻松掌握所学知识技能,达到事半功倍的效果。

精练——本套教材选材精练、详细而不冗长、简略得当,对泛泛而谈的内容将一带而过,对学生必须掌握的新技术、新方法详细讲,讲透、讲到位,为教师创造良好的教学空间和结合当地情况调整教学内容的余地。

可操作——本套教材所有的实例均是容易操作的,且是有实际意义的案例。把这些案例连接起来,就是一个应用工程的实例。通过举一反三的应用,使学生能够在更高层次上创造性地应用教材中的新思想、新技术、新方法去解决问题。

本套教材面向培养应用型人才的高等学校,同时也可作为社会培训高级技术人才的教材和需要加深某些方面知识技能的人员的自学教材。

编委会

前 言

近年来，C 语言以其功能丰富、表达能力强、使用灵活、应用面广等优点成为最流行的程序设计语言之一。目前，C 语言不仅是国内各高校普遍开设的程序设计语言课程之一，也成为学生获取计算机专业职业技术资格的初级程序员、程序员证书和参加计算机等级考试的首选语言。在高职高专的课程设置中，C 语言也被推荐为工科各专业程序设计语言的必修课程。

在本书的编写过程中，编者根据应用型高校学生的特点，结合多年讲授 C 语言的教学经验，力求做到：注重基础、强调方法、突出实用和强化实践。全书在全面阐述基础语法、语言特点的基础上，强调程序设计的方法和程序设计的风格。全书内容在满足教学大纲和计算机等级考试要求的同时，对于实际应用中 C 语言的不易掌握之处和灵活性加以重点说明。本书重视实践环节，除了每章提供上机实习题外，还有专门章节进行上机实验和课程设计指导以及程序调试方法的介绍。同时，在全书的体系结构上也作了精心编排：重点突出、难点分散、深入浅出、循序渐进。

本书由顾小晶主编，曾海、滕刚、孙浩和刘丽参编。其中第 1、2、3 章由滕刚编写，第 4、5、12 章由顾小晶编写，第 6 章的 6.1 和 6.2 以及第 10 章由孙浩编写，第 7、8、11 章由曾海编写，第 6 章的 6.3 和第 9 章由刘丽编写。沙洲工学院的顾元刚副教授在百忙中抽出时间认真细致地审阅了全书，并提出了宝贵的建议和意见，在此作者深表感谢。

在本书的编写过程中，作者得到了中国电力出版社的帮助和指导，作者所在单位的领导和同事也给予了热情帮助和支持，在此一并表示感谢。

由于作者水平有限，书中不妥之处恐在所难免，请读者批评指正。读者如有建议和要求，请发电子邮件到：clang@jssvc.com。

目 录

序 言 前 言

第 1 章 程序设计与 C 语言概述.....	1
1.1 程序设计的基本概念	1
1.2 C 语言的发展和特点	3
1.3 C 程序的基本结构	4
1.4 C 程序的上机步骤	8
习题	18
上机实习题	18
第 2 章 数据类型、运算符和表达式	20
2.1 数据类型、常量和变量	20
2.2 运算符和表达式	27
2.3 类型转换	34
习题	36
上机实习题	38
第 3 章 C 程序设计初步	39
3.1 C 语句概述	39
3.2 基本数据输入、输出方法	40
3.3 程序实例	51
习题	54
上机实习题	56
第 4 章 选择结构程序设计	57
4.1 程序设计语言的控制结构	57
4.2 if 语句	59
4.3 switch 语句	67
4.4 程序实例	69

习题	72
上机实习题	73
第 5 章 循环结构程序设计	76
5.1 while 语句.....	76
5.2 do-while 语句	77
5.3 for 语句.....	78
5.4 循环的嵌套	81
5.5 break、continue、goto 语句.....	82
5.6 程序实例	86
5.7 常用调试方法	89
习题	94
上机实习题	96
第 6 章 类别和编译预处理	97
6.1 引言	97
6.2 函数的定义	97
6.3 函数的调用和参数传递	99
6.4 变量的存储类别	102
6.5 函数的嵌套调用和递归调用.....	108
6.6 编译预处理	111
习题	117
上机实习题	118
第 7 章 指针	120
7.1 地址和指针的概念	120
7.2 指针变量	121
7.3 指针与函数	126
7.4 指向函数的指针	133
习题	137
上机实习题	138
第 8 章 数组和字符串	140
8.1 一维数组	140
8.2 二维数组	154

8.3 字符数组与字符串	160
8.4 动态分配与 void 类型指针	168
习题	171
上机实习题	172
第 9 章 结构体、共用体和枚举	173
9.1 结构体	173
9.2 结构体指针	179
9.3 共用体	188
9.4 枚举类型	191
9.5 位运算	192
习题	194
上机实习题	195
第 10 章 文件	196
10.1 文件的概述	196
10.2 缓冲文件系统	197
习题	206
上机实习题	207
第 11 章 图形设计	208
11.1 图形系统的初始化和图形模式的设定	208
11.2 屏幕颜色的设置和清屏函数	211
11.3 坐标和画笔的使用	215
11.4 TC 中的画线函数	217
11.5 TC 中的线型设定	219
11.6 填充和填充模式的设定	222
11.7 屏幕和图形窗口处理及动画	227
11.8 图形显示方式中文本的输出	231
11.9 创建独立的图形应用程序	235
11.10 实例: 综合图形设计	236
习题	238
上机实习题	238
第 12 章 上机实验和课程设计指导	239

12.1 上机实验指导	239
12.2 课程设计	240
附录 1 ASCII 代码表	247
附录 2 C 语言的关键字	251
附录 3 运算符的优先级和结合性	252
附录 4 常用 Turbo C 2.0 标准函数	253
附录 5 TC 编译、连接时的错误和警告信息	262
参考文献	

第 1 章 程序设计与 C 语言概述

1.1 程序设计的基本概念

程序设计是伴随着计算机应用发展的一门学科，是使用和开发计算机的重要工具。本节简单介绍程序设计的基本概念。

1.1.1 程序

人们要使用计算机，就必须把要解决的问题变成计算机能够接受且能执行的一条条指令，即编排一系列计算机解题步骤，这就是程序。完整地说，所谓程序，就是为完成某个任务而设计的，由有限步骤所组成的一个有机的指令序列。简言之，程序是指令的序列。程序是对所要解决问题的各个对象和处理规则的描述，是由程序员编制设计的静态过程，而输入计算机实现指令的过程是动态过程。

PASCAL 语言的创建者沃斯 (N.Wirth) 对程序给出了一个简明而确切的定义：

算法+数据结构=程序 (Algorithm+Data Structure=Program)

算法集中地反映了计算机执行的过程，数据结构是大批数据之间的构造关系，针对所处理的对象，设计好数据结构可极大地简化算法。算法和数据结构是程序的两个重要方面。早期的程序设计语言对数据结构的描述极为薄弱，造成了算法任务的加重，随着语言的发展，数据描述能力逐渐增强，这也是语言发展的一个重要标志。

程序与计算机的应用和开发密切相关，为了更好地使用计算机，必须了解程序的共同性质：

- (1) 目的性。程序必须有明确的目的，即为了解决什么问题。
- (2) 分步性。程序是分成许多步骤的，不可能一步就解决问题。
- (3) 有序性。程序不是解题步骤的随意堆积，而是要按照一定的顺序有机排列。
- (4) 有限性。解决问题的步骤不可能是无穷多，必须由有限的步骤组成，否则计算机无法实现。
- (5) 操作性。程序总是施加各种操作于某些对象，对象就是数据。

1.1.2 程序设计语言的发展和特点

程序设计语言是指用来书写计算机程序的语言，是人与计算机进行信息通信的工具。

程序设计语言目前多达上千种，常用的也有几十种。众多的程序设计语言如何进行分类，目前众说纷纭，多数人认为程序设计语言分为三大类：面向机器的语言、面向过程的语言和面向问题的语言。

(1) 面向机器的语言。面向机器的语言是针对特定的计算机而设计的语言，是不能独立于机器的语言。如机器语言和汇编语言。

(2) 面向过程的语言。面向过程的语言是用于各种计算机并能解决各种题目的语言，它是独立于机器的。使用面向过程的语言，用户不仅要告诉计算机“做什么”，而且还要告诉计算机“如何做”，需要详细地描述解题过程，因此面向过程的语言即为过程化语言。如 FORTRAN 语言、PASCAL 语言、C 语言、Ada 语言等。

(3) 面向问题的语言。面向问题的语言也是独立于计算机的语言，利用这种语言解题，不仅摆脱了计算机内部逻辑，而且不必关心问题的解法和解题的过程。只需要指出问题、输入数据和输出形式，就能得到所需要的结果。面向问题的语言与面向过程的语言之间区别就是不需要告诉计算机“如何做”，即不需要描述解题过程。因此，面向问题的语言又称为非过程化语言或陈述性语言。如报表语言、判定语言、SQL (Structured Query Language) 语言等。SQL 语言是数据库查询和操纵语言，可直接使用于数据库管理系统。非过程化语言是程序设计语言发展的目标，对用户来说是方便的，但实现起来比较困难，更为理想的非过程化语言目前仍在研制之中。

1.1.3 程序设计的一般过程

从基本概念出发，一般来说，程序设计的过程可以分为以下五个基本步骤：

1. 问题分析

问题分析是进行程序设计的基础。如果在没有把所要解决的问题分析清楚之前就编制程序，后果难于预想。基本的分析内容如下：

(1) 问题的性质。例如，你所要解决的问题是属于数值型还是属于非数值型的问题，如果是数值型的问题，就要确定一个合理的精度要求。

(2) 输入、输出数据。例如，数据的类型、输入输出的设备和格式。

(3) 数学模型或常用的方法。例如，求一元二次方程的根通常有三种方法：求根公式、韦达定理和迭代法，选择哪种方法处理结果不一样，因为计算工具的有效数字位是有限的，在运算过程中难免出现误差。

2. 问题解的描述

问题解的描述就是问题求解的步骤，是解题方案的准确而完整的描述，也就是前面所提到的算法，算法不同于一般的计算公式，应具备以下基本特征。

(1) 可行性。算法中的每一个步骤都必须是可实现的。

(2) 确定性。算法中的每一个步骤都必须有明确的定义。不允许模棱两可的解释，也不允许多义性。

(3) 有穷性。算法必须在有限的时间内完成。

(4) 一个算法应有零个或多个输入。算法可以没有输入。例如一盒音乐磁带，启动后就会放出音乐，不需要输入。

(5) 一个算法应有一个或多个输出。算法的目的是解决问题，“解”就是输出，没有输出的算法是没有意义的。

3. 编写程序

设计好合适的算法后，下一个步骤就是用选择的程序设计语言（例如 C 语言）描述该算

法, 这一工作就是编写程序, 得到的程序是文本格式的, 被称为源程序。

4. 上机调试和运行程序

计算机只能执行用机器语言编写的二进制程序(称为可执行程序), 而无法直接执行文本格式的源程序, 所以无论是用什么高级语言编写的程序, 必须翻译成机器代码才能执行。这一翻译工作由编译软件完成, 不同的语言有不同的编译软件, 而且许多编译软件还提供了源程序管理、程序调试等各种功能。在编写好程序后, 需要使用相应的编译软件上机调试和运行程序。本章的第 4 节将详细介绍如何使用 Turbo C 2.0 来调试和运行 C 程序。

5. 分析结果

程序执行后并不意味着就完成了全部工作, 还需要分析程序的输出结果, 检查其是否符合程序设计的要求, 如果出现错误, 要注意是算法的问题还是程序编写的问题, 然后不断改进, 直到得到正确的结果。

1.2 C 语言的发展和特点

1.2.1 C 语言的发展

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言。它适合于作为系统描述语言, 既可用来写系统软件, 也可用来写应用软件。C 语言的发展大致可以分为以下三个阶段:

1. C 语言的诞生 (1970—1973)

C 语言是 1972 年由美国的 Dennis M.Ritchie 设计发明的, 并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL (Basic Combind Programming Language) 发展演变而来。在 1973 年, AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的语言, 并取名为 C, C 语言就问世了。

2. C 语言的发展 (1973—1988)

为了使 UNIX 操作系统推广, 1977 年 Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本:《可移植的 C 语言编译程序》。1978 年 Brian W.Kernighian 和 Dennis M.Ritchie 出版了名著《The C Programming Language》, 从而使 C 语言成为目前世界上最为广泛流行的高级程序设计语言。

3. C 语言的成熟 (1988 至今)

随着微型计算机的日益普及, 出现了许多 C 语言版本。由于没有统一的标准, 使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况, 美国国家标准学会 (ANSI) 为 C 语言制定了一套 ANSI 标准 (称为 ANSI C 标准), 成为现行的 C 语言标准。

本书在讲解 C 语言时以 ANSI C 标准为主, 还加上了 Borland 公司 Turbo C 2.0 的一些扩充。

1.2.2 C 语言的特点

C 语言发展迅速, 而且成为最受欢迎的语言之一, 主要因为它具有强大的功能。许多著名的系统软件, 如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。用 C 语言加上一些

汇编语言子程序，就更能显示 C 语言的优势了，像 PC-DOS、WORDSTAR 等就是用这种方法编写的。归纳起来，C 语言具有下列特点：

1. 简洁、紧凑、方便、灵活

C 语言一共只有 32 个关键字，9 种控制语句，程序书写自由，主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

2. 运算符丰富

C 的运算符包含的范围很广泛，共有 34 个运算符。C 语言把括号、赋值、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. 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，可使程序效率更高。C 的数据类型有：整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂的数据类型的运算。另外 C 语言具有强大的图形功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

4. 结构化程序设计语言

结构化程序设计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、维护以及调试。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，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

5. 适用范围大，可移植性好

C 语言有一个突出的优点就是适合于多种操作系统，如 DOS、UNIX，也适用于多种机型。

上面只介绍了 C 语言的最容易理解的一般特点，至于 C 语言内部的其他特点将结合以后各章的内容作介绍。由于 C 语言具有这些优点，所以应用面很广。许多大的软件都用 C 编写，这主要是由于 C 的可移植性好和硬件控制能力高，表达和运算能力强。许多以前只能用汇编语言处理的问题，现在可以改用 C 语言来处理了。

C 的以上特点，读者现在也许还不能深刻理解，待学完 C 以后再回顾一下，就会有比较深的体会。

1.3 C 程序的基本结构

本节将介绍几个用 C 语言编写的简单程序，并通过对它们的分析来使读者了解 C 语言的特性和计算机程序设计的一般规律。

1.3.1 简单的 C 程序实例

【例 1-1】整数加法。本程序的功能是从键盘上读入两个整数，并计算这两个数的和，最



后将结果输出。

```

1  #include <stdio.h>
2  main()
3  {
4      int a,b,c;
5      printf("Enter Two Numbers:");
6      scanf("%d%d",&a,&b);
7      c=a+b;
8      printf("The Sum is %d\n",c);
9  }
```

程序运行结果为:

```

Enter Two Numbers:26 35✓
The Sum is 61
```

如果把以上的 9 行程序输入计算机后执行,就可以在显示器上得到所列的运行结果,运行结果中斜体带下划线的内容是用户通过键盘输入的(✓表示回车),其他的内容则是计算机输出的。

具体来说,程序执行后,计算机先在显示器上输出“Enter Two Numbers:”,并在文字后显示文字输入光标等待用户输入,此时用户可以输入两个整数后按回车,数之间用空格区分(如本例中的 26 35),接着计算机继续执行程序,输出结果 The Sum is 61,显然这里的 61=26+35,完全符合题目要求。

如何把 C 语言程序输入计算机并执行,将在本节后面介绍。

注意,输入的程序不包括每一行的行号,也就是每一行左侧的斜体的数字,这些行号是本书为了方便解释程序而加上的,不是程序的一部分,所以在输入程序时应该略去。

下面对照程序运行结果逐行分析每条语句的功能:

第 1 行语句的作用是将库函数 `stdio.h` 包含入本程序,如果程序中需要输入输出数据,就必须包含库函数 `stdio.h`,详细的解释将在 6.2 节中给出。

第 2、3、9 行语句一起定义了一个函数 `main`,这是 C 程序中必需的,第 4 到 8 行语句是函数体,也就是程序的具体内容。

第 4 行是变量定义语句,定义了 3 个变量 `a`、`b`、`c`,其中的 `int` 指明了这三个变量是整数类型,也就是说变量中存放的值只能是整数。本例需要处理三个整数:两个是用户输入的。一个是计算所得的,其具体的值必须要在程序执行时才能确定。在 C 语言中,这样的数据需要存放在变量中来处理,而变量又必须先定义后使用,所以这一行的语句就是用来定义变量的,变量 `a` 和 `b` 存放输入的两个整数,变量 `c` 存放 `a+b` 的运算结果。

第 5 行是输出语句,对照程序执行结果,可以看到输出的内容就是双引号内的 Enter Two Numbers,显然,如果改变这段文字,计算机就会输出不同的内容,例如要输出 Hello World!,语句就写成 `printf("Hello World!");`,`printf` 在 3.2.2 节中详细介绍。

第 6 行是输入语句,对照程序执行结果,可以看到语句中双引号内的 `%d%d` 指明要输入两个整数,后面的 `&a,&b` 则指明输入的这两个整数依次存放在变量 `a` 和 `b` 中。输入更多的内容只需要修改这两个地方,比如输入三个整数存放在变量 `x`、`y`、`z` 中的语句为 `scanf("%d%d%d",&x,&y,&z);`。

第7行语句的作用非常明显：计算 $a+b$ 的值存放在变量 c 中，在 C 语言中，算术运算加、减、乘、除分别用符号 $+$ 、 $-$ 、 $*$ 、 $/$ 表示，如果要计算变量 a 和 b 的差，可以用语句 $c=a-b$ 实现。

第8行又是一条输出语句，但对照程序执行结果，可以发现和第5行的输出语句有些不同：首先，双引号内的 $\%d$ 没有被输出，真正输出是数值 61，这正是语句中出现的变量 c 的值，所以在这里 $\%d$ 表示输出一个整数，具体的值由后面对应的变量给出。其次，双引号内的 $\backslash n$ 也没有被输出，实际输出是一个回车换行，也就是说，在 C 语言中， $\backslash n$ 表示一个回车符。

通过这个例子也可以看到，一个简单的 C 语言程序一般包括变量定义（第4行）、数据输入（第5、6行）、计算（第7行）和数据输出（第8行）四个部分。

读者可以自行改写例 1-1，以实现其他功能。例如写一个程序，输入两个整数，输出这两个数的乘积；或者输入三个整数，输出三个数的和之类的程序。

下面再举一例，让 C 语言程序带有判断能力。

【例 1-2】求最大值，输入两个整数，求两者的最大值，并输出这个最大值。

```

1    #include <stdio.h>
2    main()
3    {
4        int a,b,c;
5        printf("Enter Two Numbers:");
6        scanf("%d%d",&a,&b);
7        if(a>b)
8            c=a;
9        else
10           c=b;
11        printf("The Max is %d\n",c);
12    }
```

程序运行结果为：

```

Enter Two Numbers: 26 35✓
The Max is 35
```

将这个程序和例 1-1 做比较，可以看到两者的差别并不太大，主要的变动就是把例 1-1 中的第7行 $c=a+b$ 改成了本例中的第7、8、9、10 四行，这四行是一条 C 语言中的 if-else 语句，其作用就是比较 a 和 b 的大小，并且把较大的值保存在变量 c 中。具体来说第7行中的 $a>b$ 是 if 语句的判断条件，当条件满足时（也就是 a 大于 b 时，此时 a 为最大值），执行第8行的语句 $c=a$ （也就把最大值保存在变量 c 中了）；如果条件不满足（也就是 a 不大于 b 时，此时 b 为最大值）就执行 else 后面的第10行的语句 $c=b$ 。本书的4.2节对 if 语句有详细介绍。

同样，在大致了解例 1-2 的程序后，读者可以对其作些改动以实现其他功能，比如输入两个整数，输出其中的最小值；或者输入三个整数，输出其中的最大值。

举这两个例子，主要是为了使读者对 C 语言和计算机程序设计有一个大致的了解，具体细节不必完全弄懂，在学到以后有关章节时，问题自然迎刃而解。

1.3.2 C 程序的基本结构

通过以上几个例子，可以看到：



1. C 程序由函数构成

C 程序是由函数构成的，C 语言中用函数来实现特定的功能。一个 C 源程序至少包含一个 main 函数，也可以包含一个 main 函数和若干个其他函数。因此，函数是 C 程序的基本单位，C 的函数相当于其他语言中的子程序，可以说 C 是函数式的语言，程序中的全部工作都是由各个函数分别完成的，编写 C 程序就是编写一个个函数。此外，C 的函数库十分丰富，ANSI C 提供 100 多个库函数，Turbo C 2.0 和 MSC 4.0 提供 300 多个库函数。

C 的这种特点使得容易实现程序的模块化。

2. 函数由两部分组成

C 语言中的函数由两部分组成：

(1) 函数的首部，即函数的第一行。包括函数名、函数类型、函数属性、函数参数（形参）名、参数类型。

例 1-1 中的 main 函数的首部为

```
main ()
```

在此例中，只定义了函数名，没有给出函数的类型、参数等内容，这是允许的，但一个函数名后面必须跟一对圆括弧。

(2) 函数体，即函数首部下面的大括弧 {} 内的部分。如果一个函数内有多个大括弧，则最外层的一对 {} 为函数体的范围。

函数体一般包括：

声明部分：在这部分中定义所用到的变量，如例 1-1 中的第 3 行 “int a,b,c”。在第 6 章中还将会看到，在声明部分中要对所调用的函数进行声明。

执行部分：由若干个语句组成。

当然，在某些情况下也可以没有声明部分，甚至可以既无声明部分，也无执行部分。

3. C 程序从 main 函数开始执行

一个 C 程序总是从 main 函数开始执行的，而不论 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）。

4. 程序书写格式自由

C 程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上。C 程序没有行号，也不像 FORTRAN 或 COBOL 那样严格规定书写格式（语句必须从某一列开始书写）。

5. 语句后必须有分号

C 语言中，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，即使是程序中最后一个语句也应包含分号。

6. 程序中可以使用注释

可以用 /*.....*/ 对 C 程序中的任何部分作注释。一个好的、有使用价值的源程序都应当