

高等学校应用型规划教材 · 计算机系列



C 程序设计

实例教程

刘志海 主 编

谢凤芹 高 峰 孙 波 副主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

清华大学出版社 中国计算机教育出版集团 中国美术学院美术考级教材



C 程序设计

例题 200 例

清华大学出版社

清华大学出版社 中国计算机教育出版集团 中国美术学院美术考级教材

清华大学出版社

高等学校应用型规划教材·计算机系列

C 程序设计实例教程

刘志海 主 编

谢凤芹 高 峰 孙 波 副主编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书融理论讲解、实例分析和实验指导为一体,由浅入深,用大量不同类型实例讲解 C 程序设计方法,程序调试环境为 TurboC2.0 和 Visual C++6.0。

全书共 12 章,首先介绍了数据类型及表达式、三种基本结构的程序设计、数组、指针、结构体与链表和文件等内容;其次,特别安排综合设计实例章节,通过通讯录设计、访问 dbf 数据库、图形设计等 6 个综合实例培养读者分析问题和综合应用 C 语言基本知识解决问题的能力;最后,作为 C 语言与硬件联合应用的实例,本书还介绍了 Keil C 编译环境的应用等,以提高读者的学习兴趣,了解 C 语言的博大精深和实际应用。各章穿插大量应用实例,内容讲解透彻,程序注释完整,各章后均配有习题和编程题。

为达到更好的教学效果,本书特别为教师提供“试题库管理系统”,通过该系统可直接输出正式试卷和参考答案。此外,本书还配有电子教案、习题参考答案、源代码等教辅资料。

本书可作为高等院校公共基础课教材,也适合于工程技术人员和 C 语言自学者使用。

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 程序设计实例教程 / 刘志海主编. —北京:电子工业出版社,2008.7

高等学校应用型规划教材·计算机系列

ISBN 978-7-121-06839-3

I. C… II. 刘… 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(2008)第 080998 号

策划编辑:陈 虹

责任编辑:刘文杰 赵云红

印 刷: 北京牛山世兴印刷厂

装 订:

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: 787×1092 1/16 印张: 19.25 字数: 493 千字

印 次: 2008 年 7 月第 1 次印刷

印 数: 4 000 册 定价: 29.00 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。

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

服务热线:(010) 88258888。



前 言



C 程序设计语言最早是由 Dennis Ritchie 于 1973 年设计并实现的，从那时起，C 语言即不断展现其青春活力和卓越功能，并风靡全球，成为世界上学习和应用最多的一门高级语言，许多软件如 UNIX 操作系统、C 编译器和几乎所有的 UNIX 应用程序等，都是在 C 语言及其衍生的各种语言的基础上开发出来的。

本教材编写目的是：帮助读者学习如何运用 C 语言编写程序，提高读者对程序设计的兴趣和动手能力，使本书成为一本学习 C 程序设计合适的教材和参考书。本书在编写过程中突出了实践性教学环节，特别是对 C 程序设计方法的基本技能训练，强调培养学生在实践中应用 C 语言编程解决实际问题的能力。本书内容安排上既符合国家关于 C 语言的教学大纲的要求，又符合计算机等级考试的大纲要求。

本书是在作者近 10 年教学和实践的基础上，综合多位同行的教学科研经验精心编写而成。全书从 C 语言的语法基础入手，力求由浅入深，通过大量不同类型的实例讲解 C 语言程序的设计方法。为巩固学习成果，每章后均安排有练习题和编程题，帮助读者掌握和理解重要知识点。

全书突出特点如下：

(1) 教学辅导资料全面，除电子课件、实例源代码、习题答案等外，还包括“试题库管理系统”，教师可以直接使用该系统产生正式试卷及参考答案；

(2) 特别安排“综合实例”一章，通过 6 个综合实例，介绍了 C 语言的绘图、菜单、dbf 数据库、通讯录设计等的高级应用和综合应用，培养读者分析问题和解决实际问题的能力；

(3) 特别安排“C 语言在控制系统和嵌入式系统中的应用”一章，以提高读者兴趣、开阔视野，满足更高层次的学习需要，向 C 语言的高级应用过渡；

(4) 全书实例丰富，类型多样，内容丰富，分析透彻。既包括讲解过程中穿插的若干

程序，又特别在部分重、难点章节单独安排有小型综合实例，最后以 2 章综合应用和实验指导配合理论学习，检验效果，符合读者由浅入深接受知识的逻辑思维习惯。

对于理论教学 30 学时的专业，建议学时分配如右表所示（第 12 章为实验指导，未列入）。其他授课学时的专业，也可参照进行。

不同专业可以根据培养计划和教学大纲的要求，选讲本书第 8 章的第 8.4 节，第 10 章的第 10.3、10.4、10.5 节，第 11 章的第 11.2、11.3 节。

山东科技大学刘志海担任了全书的统稿工作，并编写了第 4、5、8、10、12 章和附录部分。高峰编写了本书的第 1、2、3 章，谢凤芹编写了本书的第 6、9、11 章，孙波参与了第 10 章的编写和校稿工作。另外，鲁青、梁慧斌参与了第 4、5 章的编写，张华宇参与了第 9、11 章的编写，王秀华参与了第 8、12 章的编写。韩以伦、李玉善在本书出版过程中给予了很多指导和帮助，部分从事 C 语言教学的同行也对本书的编写提出了许多合理的建议，在此一并表示感谢。

本书绘图部分的程序代码在 Turbo C 环境下调试通过，其余全部在 Visual C++6.0 和 Turbo C 环境下调试通过。本书配套的多媒体课件、实例源代码、习题参考答案可在电子工业出版社华信教育资源网 (<http://www.hxedu.com.cn>) 下载，或发邮件至 zhiliu@126.com 与作者联系。

由于编写时间仓促，难免存在个别不当之处，恳请读者批评指正。

30 学时	
章	学时
第 1 章	2
第 2 章	2
第 3 章	2
第 4 章	4
第 5 章	6
第 6 章	4
第 7 章	1
第 8 章	3
第 9 章	1
第 10 章	3
第 11 章	2

编 者

2008 年 4 月



目 录



第 1 章 C 语言概述	(1)
1.1 C 语言的发展	(1)
1.2 C 语言的组成及特点	(1)
1.3 C 程序的调试方法	(3)
1.4 程序设计入门	(5)
1.5 Turbo C 2.0 集成开发环境	(5)
1.5.1 Turbo C 2.0 简介	(5)
1.5.2 TurboC 2.0 集成开发环境	(6)
1.6 Visual C++ 6.0 集成环境调试	(10)
1.6.1 启动 Visual C++ 6.0	(10)
1.6.2 源程序的调试与运行	(12)
习题 1	(12)
第 2 章 数据类型	(13)
2.1 标识符和字符集	(13)
2.1.1 标识符	(13)
2.1.2 字符集	(14)
2.2 数据类型	(15)
2.3 常量和变量	(15)
2.3.1 常量和符号常量	(15)
2.3.2 变量	(16)
2.4 整型数据类型	(17)
2.4.1 整型常量	(17)
2.4.2 整型变量	(18)
2.5 浮点型数据类型	(20)

2.5.1 浮点常量	(20)
2.5.2 浮点变量	(20)
2.6 字符类型	(21)
2.6.1 字符常量	(21)
2.6.2 字符变量	(22)
2.6.3 字符串常量	(24)
2.7 不同数据类型之间的转换	(24)
2.7.1 自动转换	(24)
2.7.2 强制类型转换	(25)
习题 2	(26)
第 3 章 运算符和表达式	(27)
3.1 算术运算符和算术表达式	(27)
3.1.1 算术运算符	(27)
3.1.2 算术表达式	(27)
3.2 赋值运算符和赋值表达式	(29)
3.2.1 赋值运算符	(29)
3.2.2 赋值表达式	(29)
3.2.3 赋值语句	(30)
3.3 关系运算符和关系表达式	(31)
3.3.1 关系运算符	(31)
3.3.2 关系表达式	(31)
3.4 逻辑运算符和逻辑表达式	(32)
3.4.1 逻辑运算符	(32)
3.4.2 逻辑表达式	(32)
3.5 条件运算符和条件表达式	(33)
3.5.1 条件运算符	(33)
3.5.2 条件表达式	(34)
3.6 逗号运算符和逗号表达式	(34)
3.6.1 逗号运算符	(34)
3.6.2 逗号表达式	(35)
习题 3	(36)
第 4 章 C 语言程序的基本结构	(38)
4.1 结构化程序设计方法与算法	(38)
4.1.1 结构化程序设计方法	(38)
4.1.2 算法	(39)
4.2 C 程序语句	(44)
4.3 顺序结构程序设计	(47)
4.3.1 字符的输入和输出	(47)

4.3.2 字符串的输入与输出	(49)
4.3.3 格式化输入与输出	(50)
4.4 选择结构的基本形式	(53)
4.4.1 简单分支结构	(54)
4.4.2 双分支结构	(54)
4.4.3 多分支结构	(56)
4.4.4 switch...case 分支结构	(58)
4.5 选择结构的嵌套	(61)
4.6 循环结构的基本形式	(64)
4.6.1 if...goto...构成的循环	(64)
4.6.2 while 循环	(65)
4.6.3 do...while 循环	(67)
4.6.4 for 循环	(69)
4.7 循环结构的嵌套	(71)
4.8 循环控制语句	(73)
4.8.1 break 语句	(73)
4.8.2 continue 语句	(74)
4.9 实例	(75)
习题 4	(80)
第 5 章 数组与指针	(87)
5.1 一维数组	(87)
5.1.1 数组的定义	(87)
5.1.2 数组元素的引用	(88)
5.1.3 数组的初始化	(89)
5.1.4 一维数组的应用实例	(91)
5.2 二维数组及多维数组	(95)
5.2.1 二维数组的定义	(95)
5.2.2 二维数组的引用	(96)
5.2.3 二维数组的初始化	(97)
5.2.4 二维数组的应用实例	(98)
5.3 字符数组	(100)
5.3.1 字符数组的定义	(100)
5.3.2 字符数组的初始化	(100)
5.3.3 字符数组的引用	(101)
5.3.4 字符串	(101)
5.4 指针变量和指针运算符	(103)
5.4.1 地址与指针	(103)
5.4.2 指针变量定义及指针运算	(105)

5.4.3 指针变量的引用	(106)
5.4.4 指针的运算	(108)
5.4.5 C 语言中指针变量赋值的几种错误方法	(109)
5.5 指向数组的指针	(110)
5.5.1 指针与一维数组	(110)
5.5.2 指针与二维数组	(112)
5.5.3 指针与字符串	(114)
5.5.4 指针数组	(116)
习题 5	(119)
第 6 章 函数与参数传递	(122)
6.1 概述	(122)
6.2 函数的定义与调用	(124)
6.2.1 函数定义的一般形式	(124)
6.2.2 形式参数与实际参数	(127)
6.2.3 函数的返回值	(130)
6.2.4 函数的调用	(130)
6.2.5 函数调用时参数间的传递	(131)
6.2.6 函数的声明	(133)
6.3 函数的嵌套调用与递归调用	(134)
6.3.1 函数的嵌套调用	(134)
6.3.1 函数的递归调用	(137)
6.4 常用的数值和字符串处理函数	(141)
6.4.1 数值处理函数	(141)
6.4.2 字符串处理函数	(142)
6.5 变量的作用域和存储类型	(145)
6.5.1 局部变量	(146)
6.5.2 全局变量	(147)
6.5.3 变量的存储类别	(149)
6.6 指针作为函数的参数	(153)
6.7 指向函数的指针	(155)
6.7.1 函数的指针	(155)
6.7.2 用指向函数的指针作函数参数	(157)
6.8 返回指针的函数	(158)
6.9 main 函数中的参数	(160)
习题 6	(162)
第 7 章 编译预处理	(167)
7.1 宏定义	(167)
7.1.1 无参宏定义	(167)

7.1.2 带参宏定义	(170)
7.2 文件包含	(172)
7.3 条件编译	(173)
7.3.1 #ifdef 命令	(173)
7.3.2 #ifndef 命令	(175)
7.3.3 #if 命令	(175)
习题 7	(176)
第 8 章 结构体与链表	(177)
8.1 结构体的定义和引用	(177)
8.1.1 结构体类型定义	(177)
8.1.2 结构体类型变量的定义	(179)
8.1.3 结构体变量的初始化和成员引用	(181)
8.2 结构体数组	(183)
8.2.1 结构体数组的定义	(183)
8.2.2 结构体数组的初始化	(183)
8.3 指向结构体的指针	(186)
8.3.1 结构体指针变量的定义	(186)
8.3.2 结构体指针变量的赋值	(187)
8.3.3 结构体指针变量成员的引用	(187)
8.3.4 指向结构体数组的指针	(188)
8.3.5 结构体指针数组	(189)
8.3.6 结构体变量和结构体指针作为函数的参数	(189)
*8.4 链表的基本操作	(190)
8.4.1 单链表	(191)
8.4.2 内存操作函数	(192)
8.4.3 单链表的基本操作	(192)
8.5 共用体的定义和引用	(201)
8.5.1 共用体类型及变量的定义	(201)
8.5.2 共用体变量的引用方法	(202)
8.5.3 共用体变量的赋值	(203)
8.5.4 共用体类型数据的特点	(203)
8.6 typedef 定义类型	(205)
8.6.1 用于对数据类型的命名	(205)
8.6.2 用于对数组和指针类型的命名	(205)
8.6.3 typedef 与 #define	(206)
习题 8	(206)
第 9 章 文件	(209)
9.1 文件概述	(209)

9.1.1 文件的基本概念	(209)
9.1.2 文件的分类	(209)
9.1.3 文件的基本操作	(210)
9.2 文件类型指针	(210)
9.3 文件的打开与关闭	(211)
9.3.1 文件的打开	(212)
9.3.2 文件的关闭	(213)
9.4 文件读写	(214)
9.4.1 读字符函数 fgetc	(214)
9.4.2 写字符函数 fputc	(215)
9.4.3 写字符串函数 fputs	(217)
9.4.4 读字符串函数 fgets	(217)
9.4.5 格式化读写函数 fscanf 和 fprintf	(218)
9.4.6 数据块读写函数 fread 和 fwrite	(219)
9.5 文件的定位	(221)
9.5.1 随机定位函数 fseek()	(221)
9.5.2 文件头定位函数 rewind()	(222)
9.5.3 求当前读写位置函数 ftell()	(222)
9.6 文件检测函数	(222)
9.6.1 文件结束检测函数 feof	(222)
9.6.2 读写文件出错检测函数 ferror	(222)
9.6.3 文件出错标志和文件结束标志置 0 函数 clearerr	(223)
9.7 文件应用举例	(224)
习题 9	(226)
第 10 章 综合设计实例	(227)
10.1 万年历设计	(227)
10.1.1 功能要求	(227)
10.1.2 算法分析	(227)
10.1.3 TC2.0 工程设计	(228)
10.1.4 函数介绍	(228)
10.1.5 参考程序	(229)
10.1.6 运行结果	(230)
10.2 简单通讯录设计	(230)
10.2.1 功能要求	(230)
10.2.2 总体设计	(231)
10.2.3 存储结构	(231)
10.2.4 函数设计	(232)
10.2.5 程序示例	(233)

10.2.6 测试结果	(238)
*10.3 计算机图形设计	(238)
10.3.1 图形模式控制函数	(239)
10.3.2 图形屏幕管理函数	(240)
10.3.3 颜色与线型	(240)
10.3.4 图形操作函数	(241)
10.3.5 图形模式下文本的输出	(242)
10.3.6 实例	(243)
10.3.6 程序示例	(244)
*10.4 图形模式下菜单设计	(245)
10.4.1 菜单设计要点	(246)
10.4.2 下拉菜单设计实例	(247)
*10.5 Windows 下字模转换	(255)
10.5.1 国标汉字字符集与区位码	(256)
10.5.2 汉字的内码	(256)
10.5.3 内码转换为区位码	(257)
10.5.4 实例	(257)
10.6 读取 dbf 数据表格	(259)
10.6.1 dbf 表格文件的结构	(260)
10.6.2 读取 dbf 表格的内容	(261)
习题 10	(263)
第 11 章 C 语言在控制系统和嵌入式系统中的应用	(264)
11.1 位运算符	(264)
11.1.1 “按位与”运算符 (&)	(265)
11.1.2 “按位或”运算符 ()	(266)
11.1.3 “按位异或”运算符 (^)	(266)
11.1.4 “求反”运算符 (~)	(267)
11.1.5 “左移”运算符 (<<)	(268)
11.1.6 “右移”运算符 (>>)	(268)
11.1.7 位复合赋值运算符	(268)
*11.2 嵌入式系统 C 语言编程实例	(269)
*11.3 MCS51 单片机的 Keil C 环境	(272)
习题 11	(274)
第 12 章 实验指导	(275)
实验一 TurboC2.0 和 VC6.0 集成环境调试	(275)
实验二 数据类型、运算符和表达式	(276)
实验三 顺序程序设计	(278)
实验四 选择结构程序设计	(278)

实验五 循环控制	(279)
实验六 数组	(280)
实验七 指针	(281)
实验八 函数	(282)
实验九 预处理与指针	(283)
实验十 结构体与共同体	(284)
实验十一 文件、位操作	(284)
实验十二 综合程序设计	(285)
附录 A 运算符的优先级	(287)
附录 B 常用字符与 ASCII 代码对照表	(289)
附录 C 常见调试错误	(291)
参考文献	(294)

第 1 章 C 语言概述

【本章概述】

C 语言是一种通用的程序设计语言，尤其适合于编写编译器和操作系统。C 语言除提供很多数据类型之外，还提供了基本的控制流结构，如选择结构、循环结构等。在深入学习和掌握利用 C 语言进行程序设计之前，有必要了解 C 程序文件的组成，以及 C 语言程序的编译和调试过程。

【学习要求】

- 了解：C 语言的发展
- 掌握：C 语言的特点、简单 C 程序的组成
- 掌握：C 程序的上机调试步骤
- 重点：简单 C 程序的组成、上机调试步骤
- 难点：集成开发环境的学习、程序调试方法

1.1 C 语言的发展

C 语言是世界上最流行的计算机程序设计语言之一。

C 语言是在 1978 年由美国电话电报公司（AT&T）贝尔实验室正式发表的，C 语言的许多重要概念来源于由 Martin Richards 开发的 BCPL 语言，BCPL 对 C 语言的影响间接来自于 B 语言，它是 1970 年 Ken Thompson 在 DEC PDP-7 计算机上为第一个 UNIX 系统开发的。由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《The C Programming Language》一书，奠定了 C 语言的基础，形成了 K&R 的 C 语言标准。但是，在《The C Programming Language》中并没有定义完整的 C 语言标准。1983 年，由美国国家标准学会（ANSI）在此基础上制定了 C 语言标准，通常称之为 ANSI C。1987 年，ANSI 又公布了新标准——87 ANSI C。1990 年，国际标准化组织（ISO）接受 87 ANSI C 为 ISO C 的标准（ISO9899-1990），目前流行的 C 编译系统都是以该标准为基础的。

1.2 C 语言的组成及特点

C 语言由于它的强大功能和诸多优点逐渐为人们所知，并很快在各类大、中、小和微型计算机上得到了广泛地使用，成为当代最优秀的程序设计语言之一，并形成多种版本，目前最流行的有 Microsoft C（或称 MS C），Borland Turbo C（或称 Turbo C），AT&T C。

这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自做了一些扩充，使之

更加方便、完美。

C 语言是一种结构化程序设计语言，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，并且 C 语言的表现能力和处理能力极强。其主要特点如下所述。

(1) 简洁、紧凑，使用方便、灵活。C 语言只有 32 个关键字、5 种基本语句、9 种控制语句，程序书写形式自由，主要用小写字母表示。

(2) 具有丰富的运算符和数据类型，便于实现各类复杂的数据结构。C 语言提供的数据类型有整形、实型、字符型、数组类型、指针类型、结构体类型和共用体类型等。特别是指针类型，使得程序员能够通过操作内存空间地址来直接处理数据，提高了程序设计灵活性及执行效率。

(3) 能够直接访问内存的物理地址，进行位 (bit) 一级的操作。具有汇编语言的部分功能，能直接对硬件进行操作，因此 C 语言又被称为“中级语言”。

(4) 具有结构化控制语句，便于实现程序的模块化设计。

(5) 既可用于系统软件的开发，也适合于应用软件的开发。

(6) C 语言编制的程序较其他高级语言编制的程序更具有效率高、可移植性强等特点。下面通过一个简单的小程序开始学习 C 语言。

【例 1.1】 一个简单的小程序。

```
#include <stdio.h>           /*编译预处理包含命令*/
void main()                  /*主函数*/
{
    printf("hello,world! "); /*格式输出函数*/
}
```

上述程序即称为 C 语言源程序，简称 C 程序。

该程序第一行中 `include` 是一条编译预处理命令，其含义是把尖括号(<>)或引号(")内指定的文件包含到本程序中，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各个标准库函数的函数原型。因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件。如例 1.1 中，`include` 的作用是将后面的 `stdio.h` 头文件包含到设计的程序中来，因为在 `stdio.h` 头文件中有程序要用到的 `printf` 函数。

`main` 是一个函数名，表示“主函数”。C 程序总是由一个或多个函数组成，程序通过函数实现要做的各种操作，函数名可以按照标识符的命名法则随程序员的喜好去取，但是需要注意的是在 C 程序中“主函数”只有一个，就是 `main` 函数，C 程序总是从 `main` 函数中的第一条语句开始执行，至主函数中的最后一条语句结束运行。

花括号({})括起来的部分是函数的语句部分，称为函数体。例 1.1 的函数体中只有一条语句：

```
printf("hello,world! ");
```

这是一条函数调用语句，`printf` 是函数名称，它的定义在 `stdio.h` 头文件中。该语句的功能是将输出""中的内容。在 C 语言中，语句以“;”作为结束符。

注意：`printf("hello,world! ")`只是函数调用，只有后面加上“;”之后才能构成一条 C 语言的语句。

`/*...*/`中间的部分是注释，是为了便于程序阅读及维护而添加的，对于程序的编译和

执行没有影响。注释可以加到程序的任何位置，但需要注意的是注释不能够嵌套，即在注释中不能再含有/*...*/。

上面程序在 Turbo C 2.0 编译系统（使用方法见附录）中经过编辑、编译、连接和执行之后的运行结果是在显示器中输出字符串：

```
Hello,world!
```

【例 1.2】C 语言程序实例。

```
#include<math.h>
#include<stdio.h>
void main()
{
    double x,s;                /*变量定义*/
    printf("input number:\n"); /*格式输出函数*/
    scanf("%lf",&x);           /*格式输入函数，定义在 stdio.h 头文件中*/
    s=sin(x);                  /*正弦函数，定义在 math.h 头文件中*/
    printf("sine of %lf is %lf\n",x,s);
}
```

通过前面例 1.1 的学习，可以看出在例 1.2 中有两条 include 命令，其作用是分别将 math.h、stdio.h 两个头文件包含进来。

在本例主函数体中又分为两部分，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。在本例函数体的 5 条语句中，double x,s; 为函数体的说明部分，是变量定义语句，该语句的作用是通过编译程序告诉计算机在存储空间内划分出两个 double 型的空间提供给变量 x 和 s 使用，分别表示输入的自变量和 sin 函数值。因例 1.1 中未使用任何变量，所以无说明部分。

注意：在 C 语言中规定，源程序中所有用到的变量都必须先说明、后使用，否则会出错。

函数体中变量定义语句后面的 4 条语句是执行部分或称为执行语句部分，用以完成程序的指定功能。执行部分的第 1 行是输出语句，调用 printf 函数在显示器上输出提示字符串，提示操作人员输入自变量 x 的值；第 2 行为输入语句，调用 scanf 函数，接受键盘上输入的数并存入变量 x 中；第 3 行是调用 sin 函数并把函数值赋值给变量 s；第 4 行是用 printf 函数输出变量 s 的值，即 x 的正弦值。

将本程序编译、连接和执行后，首先在显示器上输出提示串 input number，这是由执行部分的第一个 printf 函数完成的；然后从键盘上键入某一数值，如 30，按回车键，接着在屏幕上给出计算结果，运行结果如下：

```
input number:
30
sine of 30.000000 is -0.988032
```

1.3 C 程序的调试方法

编写一个 C 语言程序到完成运行的基本过程如图 1-1 所示。表 1-1 对程序设计各阶段