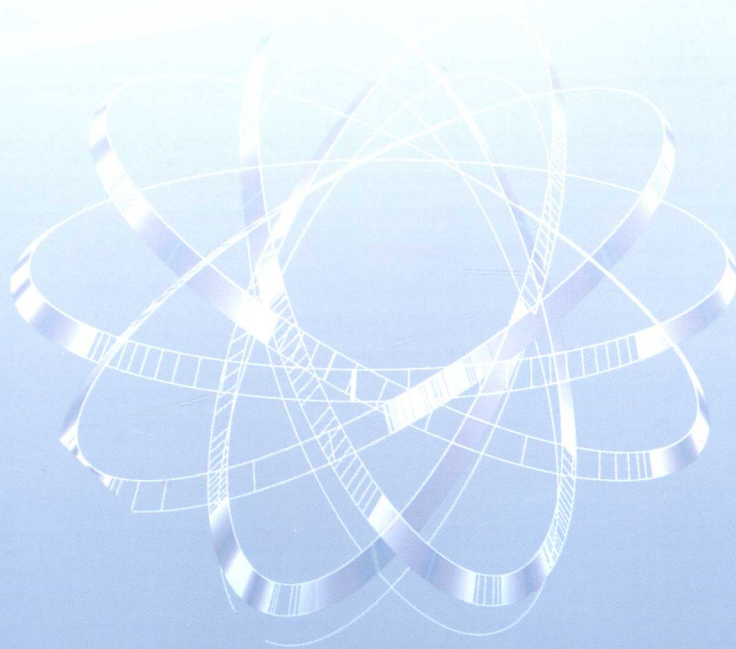


▶ 普通高等教育“十一五”规划教材 ●●●●●

C程序设计实用教程

张建勋 纪 钢 主 编
陈 渝 刘春萌 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等教育“十一五”规划教材

C 程序设计实用教程

主 编 张建勋 纪 钢

副主编 陈 渝 刘春荫

图书在版编目(CIP)数据

C 程序设计实用教程 / 张建勋, 纪钢主编. — 北京: 中国铁道出版社, 2009.1

ISBN 978-7-113-08870-8

I. C… II. ①张… ②纪… III. C 程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2008)第002550号

书 名: C 程序设计实用教程

作 者: 张建勋 纪钢 主编

责任编辑: 严晓梅

封面设计: 王 芳

版式设计: 张 颖

封面设计: 付 彪

责任印制: 李 佳

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

C 语言功能丰富,表达力强,使用灵活方便,应用面广,目标程序高,可移植性好,兼具高级语言和低级语言的许多优点。高校的高级语言程序设计课程主要以 C 语言作为程序设计语言。全书共分 11 章,内容包括:C 语言程序设计基础、基本数据类型、运算符与表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、指针、编译预处理命令、复杂数据类型、文件。

本书注重教材的可读性和适用性,每章开头都给出了双语教学中的英语词汇;在“承上启下”中介绍本章内容与前后章节知识点的关系;在“指点要律”中给出该章学习意义和学习目标;在“小结”中对关键知识点和要求掌握的知识点进行了详细的说明;附有大量的图表、程序使读者能正确、直观地理解问题;样例程序由浅入深,用于强化知识点、算法、编程方法与技巧,并给出了详细的解释;另外,本书还配套提供题型丰富的习题。

本书适合作为高等院校本科“C 程序设计”课程的教学用书,同时也可作为高职高专相关专业教学用书,还可作为全国计算机等级考试及各类培训班的培训教材。

图书在版编目(CIP)数据

C 程序设计实用教程/张建勋,纪钢主编. —北京:中国铁道出版社,2009.1

普通高等教育“十一五”规划教材

ISBN 978-7-113-08879-8

I. C… II. ①张… ②纪… 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(2009)第 005260 号

书 名: C 程序设计实用教程
作 者: 张建勋 纪 钢 主编

策划编辑: 严晓舟 吕燕新

责任编辑: 王雪飞

编辑助理: 张国成

封面设计: 付 巍

责任印制: 李 佳

编辑部电话: (010) 63583215

封面制作: 白 雪

出版发行: 中国铁道出版社(北京市宣武区右安门西街 8 号)

邮政编码: 100054)

印 刷: 北京市彩桥印刷有限责任公司

版 次: 2009 年 1 月第 1 版 2009 年 1 月第 1 次印刷

开 本: 787mm×1092mm 印张: 18 字数: 447 千

印 数: 3 000 册

书 号: ISBN 978-7-113-08879-8/TP·2890

定 价: 29.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前言

FOREWORD

C 语言是当今最流行的程序设计语言之一，它功能丰富，表达力强，使用灵活方便，应用面广，目标程序高，可移植性好，兼具有高级语言和低级语言的许多优点，适合作为系统描述语言，既可以用来编写系统软件，也可以用来编写应用软件。C 语言诞生后，许多原来用汇编语言编写的软件就可以用 C 语言编写了（例如，著名的 UNIX 操作系统就是用 C 语言编写的），而学习和使用 C 语言要比学习和使用汇编语言容易得多。因此，高校的高级语言程序设计课程主要以 C 语言作为程序设计语言。

C 语言程序设计是计算机专业的入门课程，该课程不仅重视理论教学，更强调学生的动手实践能力，迫切需要一本内容新颖、体系合理、逻辑性强、文字流畅、通俗易懂的学习 C 语言的理想教材，使具有计算机初步知识的读者都能读懂该书。实践性教学方面需要案例教学、启发式教学等先进教学手段设计的验证性、综合性、创新性实验。

本书注重教材的可读性和适用性，每章开头都给出了双语教学英语词汇，在“承上启下”中介绍本章内容与前面章节知识点的关系、本章内容与后面章节知识点的关系；在“指点要律”中给出该章学习意义和学习目标；在“小结”中对关键知识点和要求掌握的知识点进行了详细的说明；附有大量的图表、程序使读者能正确、直观地理解问题；样例程序由浅入深，强化知识点、算法、编程方法与技巧，并给出了详细的解释；另外，本书还配套提供题型丰富的习题。目录中加“*”的章节为选修内容。

全书由重庆工学院“C 程序设计”精品课程建设小组的教师集体编写完成。本书的作者都是长期在高校从事“C 程序设计”教学的一线教师，有丰富的教学经验和软件开发能力，全书文字流畅、通俗易懂、概念清楚、深入浅出、例题丰富、实用性强。

本书由张建勋、纪钢担任主编，由陈渝、刘春萌担任副主编。第 1 章、第 2 章由崔建国博士编写，第 3 章由金艳编写，第 4 章由刘春萌副教授编写，第 5 章由付国瑜编写，第 6 章由张建勋教授编写，第 7 章由杨长辉编写，第 8 章由纪钢教授编写，第 9 章由洪雄编写，第 10 章由王华秋博士编写，第 11 章由陈渝副教授编写。全书由张建勋教授统稿。

本书适合作为高等院校本科“C 程序设计”课程的教学用书，同时也可作为高职高专院校相关专业的教学用书，还也可作为全国计算机等级考试及各类培训班的培训教材。

为了方便教学，本书配套提供教师用电子教案和精品课程“C 语言程序设计”教学网站材料，另外，我们还开发了无纸化的计算机网络考试系统，使用本书的院校可通过电子邮件 zhangjianxuncq@sina.com 向作者索取。

由于我们水平有限，本书可能会有不尽人意之处，疏漏在所难免，敬请读者批评指正，以便我们及时修正。

编 者

2008 年 10 月

目 录

CONTENTS

第 1 章 C 语言程序设计基础	1
1.1 C 语言的发展及其特点和应用	3
1.1.1 C 语言的发展	3
1.1.2 C 语言的特点	3
1.2 C 语言程序的基本结构	4
1.3 算法表示方法	7
1.4 运行 C 语言程序的基本步骤	12
小结	15
习题一	15
第 2 章 基本数据类型、运算符与表达式	18
2.1 C 语言的数据类型	19
2.2 常量、变量和标识符	20
2.3 简单数据类型与表示范围	23
2.3.1 整型数据	23
2.3.2 实型数据	26
2.3.3 字符型数据和字符串常量	28
2.3.4 各数据类型间的混合运算	31
2.3.5 简单数据类型的表示范围	33
2.3.6 数据的简单输出	33
2.4 C 语言的运算符与表达式	34
2.4.1 C 语言运算符简介	34
2.4.2 赋值运算符和赋值表达式	36
2.4.3 强制类型转换符	40
2.4.4 算术运算符和算术表达式	41
2.4.5 自增自减运算符	42
2.4.6 位运算符、位运算表达式	44
2.4.7 逗号运算符和逗号表达式	46

2.4.8 指针运算符、sizeof 运算符	47
2.5 运算符的优先级和结合性	48
小结	48
习题二	50
第 3 章 顺序结构程序设计	53
3.1 结构化程序设计	54
3.1.1 结构化程序设计概述	54
3.1.2 结构化程序设计的基本结构及其特点	54
3.2 算法	55
3.2.1 算法的基本概念	55
3.2.2 算法的特性	55
3.2.3 算法的流程图表示法	56
3.2.4 基本算法	57
3.3 C 语句概述	58
3.4 输入/输出语句	59
3.4.1 格式化输出函数 printf()	60
3.4.2 格式化输入函数 scanf()	62
3.4.3 字符输出函数 putchar()	63
3.4.4 字符输入函数 getchar()	64
3.5 顺序结构程序设计举例	65
小结	66
习题三	67
第 4 章 选择结构程序设计	69
4.1 C 语言程序中语句的分类	69
4.2 关系运算符、逻辑运算符、条件运算符	71
4.2.1 关系运算符和关系表达式	71
4.2.2 逻辑运算符和逻辑表达式	72
4.2.3 条件运算符和条件表达式	73
4.3 选择结构的程序设计	73
4.3.1 if 语句	73
4.3.2 switch 语句	74
4.4 选择结构程序设计举例	76
小结	79
习题四	79
第 5 章 循环结构程序设计	81
5.1 while 语句	82

5.2 do...while 语句	84
5.3 for 语句	86
5.4 goto、break、continue 语句	88
5.6 循环的嵌套	90
5.7 程序举例	91
小结	93
习题五	94
第 6 章 数组	95
6.1 一维数组的定义和引用	97
6.1.1 一维数组的定义	97
6.1.2 一维数组元素的引用	97
6.1.3 一维数组的初始化	98
6.1.4 一维数组程序举例	99
6.2 二维数组	101
6.2.1 二维数组的定义	101
6.2.2 二维数组元素的引用	102
6.2.3 二维数组的初始化	102
6.2.4 二维数组程序举例	103
6.3 字符数组	106
6.3.1 字符数组的定义	106
6.3.2 字符数组的初始化	106
6.3.3 字符数组的引用	107
6.3.4 字符串	108
6.3.5 字符串的输入/输出	108
6.3.6 字符串操作函数	111
6.3.7 字符数组应用举例	113
小结	115
习题六	116
第 7 章 函数	117
7.1 函数概述	120
7.1.1 函数与模块化程序设计方法	120
7.1.2 函数的分类	120
7.1.3 主函数	121
7.2 函数定义和调用	121
7.2.1 函数定义	122
7.2.2 函数调用	125
7.3 函数参数传递	126

7.3.1 传值调用	126
7.3.2 传址调用	127
7.4 函数与数组	129
7.5 函数与指针	133
7.5.1 返回指针的函数	134
*7.5.2 指向函数的指针	135
7.6 函数与结构	136
7.6.1 结构指针及结构变量的传址调用	137
7.6.2 结构型函数	138
7.6.3 结构指针型函数	139
7.7 递归函数	140
*7.8 命令行参数	144
7.9 标准库函数	146
7.10 程序举例	147
小结	151
习题七	151
第 8 章 指针	154
8.1 指针与指针变量的概念	155
8.1.1 指针的概念	155
8.1.2 指针变量	155
8.2 指针变量的定义和引用	157
8.2.1 指针变量的定义	157
8.2.2 指针变量的引用	158
8.3 指针和地址运算	159
8.4 指针与数组	161
8.4.1 指针变量与数组	161
8.4.2 指针变量在一维数组中的应用	163
8.4.3 指针变量在多维数组中的应用	165
8.5 指针与字符串	169
8.5.1 字符串操作的特点及字符指针变量的引入	169
8.5.2 指向字符串的指针变量	170
8.6 指针数组与多级指针	172
8.6.1 指针数组的定义	172
8.6.2 指针数组的使用	173
8.6.3 多级指针	174
8.7 指针变量与函数	175
8.7.1 函数的操作方式与指针变量	175

8.7.2	指针型函数的定义与使用	176
8.7.3	函数指针的定义与使用	176
8.7.4	与指针有关的函数参数传递方式	178
8.7.5	带参数的 main 函数和命令行参数	183
8.8	指针与动态内存分配	184
8.8.1	动态存储的概念	184
8.8.2	C 语言的动态存储管理方式	184
	小结	186
	习题八	188
第 9 章	编译预处理	194
9.1	宏定义	195
9.1.1	不带参数的宏定义	196
9.1.2	带参数的宏定义	197
9.2	文件包含处理	200
9.3	条件编译	204
	小结	207
	习题九	209
第 10 章	复杂数据类型	211
10.1	复杂数据类型概述	212
10.2	结构体	213
10.2.1	结构体类型的定义	213
10.2.2	结构体变量的定义和引用	213
10.2.3	结构体变量的赋值	216
10.2.4	结构体数组	217
10.3	线性链表	220
10.3.1	固定内存分配与动态内存分配的概念	220
10.3.2	链表的概念	221
10.3.3	动态存储分配链表的基本操作	223
10.4	共用体	226
10.4.1	共用体类型的定义	226
10.4.2	共用体变量的定义和引用	226
10.4.3	共用体变量的赋值	228
10.5	枚举类型	228
10.5.1	枚举类型的定义	228
10.5.2	枚举变量的说明与引用	229
10.6	自定义类型	230
10.7	复杂数据类型应用综合举例	231

小结	234
习题十	235
第 11 章 文件	240
11.1 文件的基本概念	241
11.2 文件的类别	242
11.3 文件指针	243
11.4 文件操作概述	244
11.5 文件的打开、读/写和关闭	245
11.5.1 文件的打开 (fopen 函数)	245
11.5.2 文件的关闭 (fclose 函数)	246
11.5.3 文件的读/写	247
11.6 文件的定位读/写	252
11.7 文件应用综合举例	254
小结	257
习题十一	257
附录 1 C 语言的字符集——ASCII 字符表	258
附录 2 C 语言的关键字	259
附录 3 运算符的结合性	260
附录 4 C 语言的库函数	261
附录 5 C 语言常见错误	268
附录 6 Turbo C 常见错误提示英汉对照	272
参考文献	278

第 1 章 | C 语言程序设计基础

本章双语教学英语词汇

保留字	reserved words	连接	link
编辑	edit	流程图	flow chart
编码	code, encode	面向对象程序设计	object oriented programming
编译	compile	模块化	modular
运行	run	结构化	structured
标识符	identifier	操作符	operator, actor
目标程序	object program	嵌套	nest
常量	constant	软件设计	software design
定义	define	数据结构	data structure
二进制	binary	顺序结构	sequential structure
分隔符	separator	算法	algorithm
分支结构	branch construct	循环结构	loop structure
符号	symbol	运行环境	operational environment
微型计算机	micro computer	个人计算机	personal computer
信息	information	工作区	work area
序列	sequence	关键字	keyword
头文件	header file	函数	function
语句	statement	函数体	function body
源程序	source program	汇编语言	assembly language
机器语言	machine language	高级语言	higher language
自然语言	human language	致命错误	error
解释	interpretation	注释	explanatory note, comment
警告	warning	空语句	dummy statement

本章学习指导（概述）

承上启下

本章内容与后面章节知识点的关系：

本章主要介绍 C 语言的发展简史、特点、基本结构和 C 程序的编辑、编译、运行方法，以及

阅读简单的 C 语言程序的方法。在了解算法的各种表示方法的基础上,掌握算法的设计及流程图的表示方法,从总体上把握 C 语言程序的结构、格式和风格。

本章内容与下一章节的知识点相关。主要有:

1. C 程序的运行过程

C 语言程序运行过程的编辑、编译、连接和执行步骤,以及源程序、目标程序和可执行程序的概念。

2. C 语言程序的基本结构

(1) 一个 C 语言源程序可以由一个或多个源文件组成。

(2) 每个源文件可由一个或多个函数组成。

(3) 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。

(4) 源程序中可以有预处理命令 (`include` 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。

(5) 每一个说明,每一个语句都必须以分号结尾。

(6) C 语言用函数进行输入/输出,如 `printf()`、`scanf()`。

(7) C 语言用 `/**/` 或 `//` 作注释。

(8) C 程序书写格式自由,一个语句可以占多行,一行也可以有多个语句。

3. 算法的表示方法

流程图的画法。

指点要律

1. C 语言的特点

(1) C 语言简洁、紧凑,使用方便、灵活。

(2) 运算符丰富。

(3) 数据结构类型丰富。

(4) C 语言是一种结构化、模块化的程序设计语言。

(5) 语法限制不太严格,程序设计自由度大。

(6) C 语言既具有高级语言的功能,又具有汇编语言的许多功能。

(7) 生成目标代码质量高、程序执行效率高,与汇编语言相比,用 C 语言编写的程序可移植性好。

2. C 语言程序结构特点

(1) 一个 C 语言源程序可以由一个或多个源文件组成。

(2) 每个源文件可由一个或多个函数组成。

(3) 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。

(4) 源程序中可以有预处理命令 (`include` 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。

(5) 每一个说明,每一个语句都必须以分号结尾。

(6) C 语言用函数进行输入输出,如 `printf()`、`scanf()`。

(7) C 语言用 `/**/` 或 `//` 作注释。

(8) C 程序书写格式自由,一个语句可以占多行,一行也可以有多个语句。

3. C 语言程序的运行过程

C 语言程序运行过程的编辑、编译、连接和执行步骤,以及源程序、目标程序和可执行程序的概念。

4. 算法的表示方法

为解决一个问题而采取的方法和步骤,就称为算法。算法的表示方法有很多种,常用的有自然语言描述、伪代码、流程图、N-S图等。重点要求掌握流程图和N-S图的画法。

1.1 C语言的发展及其特点和应用

1.1.1 C语言的发展

C语言是当今最流行的程序设计语言之一,它适合作为系统描述语言,既可以用来编写系统软件,也可以用来编写应用软件。

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ALGOL60。ALGOL60出现于1960年,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。随后,出现了BCPL(Basic Combined Programming Language)语言。1970年,美国贝尔实验室出现了很简单而且很接近硬件的B语言(取BCPL的第一个字母)。1972—1973年,美国贝尔实验室的D.M.Ritchie在B语言的基础上设计出来C语言(取BCPL的第二个字母),并用它改写了UNIX(UNIX第五版)。C语言和UNIX可以说是一对孪生兄弟,在发展过程中相辅相成。1978年,由美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。同时由B.W.Kernighan和D.M.Ritchie合著了著名的《THE C PROGRAMMING LANGUAGE》一书。后来由美国国家标准协会(American National Standards Institute)在此基础上制定了一个C语言标准,通常称之为ANSI C。

C语言在推广过程中,出现了多种版本,目前最流行的C语言有以下几种:

- 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作了一些扩充,使之更加方便、完美。

本书所有示例均在Turbo C 2.0或Turbo C++ 3.0环境下通过调试。

1.1.2 C语言的特点

C语言不仅具有高级语言的特点,并且还具有汇编语言的功能,同时具有极强的兼容性和良好的用户界面。C语言的主要特点如下:

(1) C语言简洁、紧凑,使用方便、灵活。ANSI C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,压缩了一切不必要的成分。注意: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、逗号等都作为运算符处理,从而使C的运算类型极为丰富,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。具有当代一般高级语言的各种数据结构,又具有特别的指针类型。

(4) C语言是一种结构化、模块化的程序设计语言。其程序由函数组成,便于模块化的程序设计,程序结构完全由顺序结构、选择结构和循环结构组成。C语言通过9种结构控制语句来描述各种结构。

(5) 语法限制不太严格, 程序设计自由度大。一行可以写多个语句, 变量类型使用灵活。

(6) C 语言既具有高级语言的功能, 又具有汇编语言的许多功能。它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, 有人把它称为中级语言。

(7) 生成目标代码质量高, 程序执行效率高, 与汇编语言相比, 用 C 语言编写的程序可移植性好。C 语言一般只比汇编程序生成的目标代码效率低 10%~20%, 基本上不用修改就可以用于各种计算机系统。

1.2 C 语言程序的基本结构

1. 简单的 C 程序

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, 但可从这些例子中就可以了解到组成一个 C 语言源程序的基本部分和书写格式。

【例 1.1】输出指定字符串。

(1-1.cpp)

```
#include<stdio.h>
void main()
{
    printf("Hello, this is a c programming language.\n");
    /*输出 C 程序语言字符串*/
}
```

程序运行结果:

Hello, this is a c programming language.

程序分析:

(1) C 语言程序一般用小写字母书写。

(2) 每个 C 语言源程序必须要有且只能一个 main() 函数, 称主函数; main 前的 void 表示此函数是“空类型”, void 是“空”的意思, 即执行此函数后不产生一个函数值。

注: C 语言程序是由函数构成的, 函数是 C 语言程序的基本单位。函数有系统提供的库函数和用户设计的函数。

(3) 程序体必须在一对 { } 之间。

(4) 每个语句的结尾, 必须要有“;”作为终止符。

(5) /*……*/ 表示注释部分, 注释内容可以用汉字或英文字符表示。注释只是给人看的, 对编译和运行不起作用。注释可以出现在一行中的最右侧, 也可以单独成为一行, 可以根据需要写在程序的一行中。

(6) 函数调用语句, printf 函数的功能是把要输出的内容送到显示器去显示。

(7) printf 函数是一个由系统定义的标准函数库中的输出函数, 可在程序中直接调用。printf 语句中双撇号中的字符串按原样输出。“\n”是换行符, 即在输出“Hello, this is a c programming language.”后回车换行。

(8) #include 为预编译命令, 也称文件包含命令, 一般位于 main 主函数之前几行, 用于将有关的“头文件”包括到用户源文件中。其意义是把尖括号<>或引号""内指定的文件包含到本程序

来,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。扩展名为.h的文件称为头文件,“stdio.h”为标准输入/输出库文件,在该文件定义了 printf 函数的原型。stdio 是“standard input&output”的缩写,即有关标准输入/输出的信息。

【例 1.2】计算指定函数关系式。

(1-2.cpp)

```
#include<math.h>
#include<stdio.h>
void main()
{
    float x,y;                /*定义两实型变量*/
    printf("input number:\n");
    scanf("%f",&x);           /*输入数字*/
    y=2*sqrt(x)+1;            /*计算指定函数关系式的结果*/
    printf("2*sqrt(%f)+1=%f\n",x,y); //按制定格式输出
}
```

程序运行结果:

```
input number:
9↵
2*sqrt(9.000000)+1=7.000000
```

程序分析:

程序的功能是从键盘输入一个实数 x ,求 2 倍开方 x 加 1 关系式的值,然后输出结果。在 main() 之前的两行都是预处理命令。

(1) 凡是在程序中调用一个库函数时,都必须用预处理命令调用该函数原型所在的头文件。在本例中,使用了 3 个库函数:输入函数 scanf,开平方函数 sqrt,输出函数 printf。sqrt 函数是数学函数,其头文件为 math.h,因此在程序的主函数前用 include 命令包含了 math.h。scanf 和 printf 是标准输入/输出函数,其头文件为 stdio.h,在主函数前也用 include 命令包含了 stdio.h 文件。

(2) /*.....*/ 注释语句可以出现在一行中的最右侧,也可以单独成为一行,可以根据需要写在程序的一行中。注释也可以用“//”符号标识注释内容,这种注释只在一行有效,在执行语句的前面和中间不能使用这种注释。

(3) 例题中的主函数体又分为两部分,一部分为说明部分,另一部为执行部分。说明是指变量的类型说明。例题 1.1 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定义,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。本例中使用了两个变量 x 、 y ,用来表示输入的自变量和 sqrt 函数值。说明部分后的四行为执行部分或称为执行语句部分,用以完成程序的功能。执行部分的第一行是输出语句,调用 printf 函数在显示器上输出提示字符串,请操作人员输入自变量 x 的值。第二行为输入语句,调用 scanf 函数,接收键盘上输入的数并存入变量 x 中。第三行是调用 sqrt 函数并把函数值送到变量 y 中。第四行是用 printf 函数输出变量 y 的值,即 $2*\sqrt{x}+1$ 的正弦值,然后程序结束。

(4) 运行本程序时,首先在显示器屏幕上给出提示串 input number,这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数,如 9,按下回车键(Enter),接着在屏幕上给出计算结果。

在前两个例子中用到了输入和输出函数 scanf 和 printf,为了便于理解。这里先简单介绍它们的格式,以便下面使用。

`scanf` 和 `printf` 这两个函数分别称为格式输入函数和格式输出函数，其意义是按指定的格式输入/输出值。这两个函数的一般形式为：

`printf` 或 `scanf` (格式控制, 参数表)

格式控制是一个字符串，必须用双引号括起来，它表示了输入/输出量的数据类型。各种类型的格式表示法可参阅后续章节。在 `printf` 函数中还可以在格式控制内出现非格式控制字符，这时在显示屏幕上将按原样输出。参数表中给出了输入或输出的量。当有多个量时，用逗号间隔。例如：

```
printf("2*sqrt(%f)+1=%f\n",x,y);
```

其中 `%f` 为格式字符，表示按实数处理。它在格式串中两次出现，分别对应了 `x` 和 `y` 两个变量。其余字符为非格式字符则按原样输出在屏幕上。

【例 1.3】 求两个数的最大值。

(1-3.cpp)

```
#include<stdio.h>
int max(int x,int y)                /*定义 max 函数*/
{
    int m;
    if(x>y)
        m=x;
    else
        m=y;
    return m;                       /*把结果返回主调函数*/
}
void main()                        /*主函数入口，程序执行起点*/
{
    int a,b,c;                     /*变量说明*/
    int max(int a,int b);          /*函数说明*/
    printf("input two numbers:a,b\n");
    scanf("%d,%d",&a,&b);          /*输入 a,b 的值*/
    c=max(a,b);                    /*调用 max 函数*/
    printf("max=%d\n",c);          /*输出两数的最大值*/
}
```

程序运行结果：

```
input two numbers:a,b
5,9↵
max=9
```

程序分析：

例 1.3 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大的数。

本程序由两个函数组成，主函数和 `max` 函数。函数之间是并列关系。可从主函数中调用其他函数。`max` 函数的功能是比较两个数的大小，然后把较大的数返回给主函数。`max` 函数是一个用户自定义函数，其返回值类型是整型。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。在程序的每行后用 `/*` 和 `*/` 括起来的内容为注释部分，程序不执行注释部分。

例 1.3 中程序的执行过程是，首先在屏幕上显示提示字符串，请用户输入两个数，回车后由 `scanf` 函数语句接收这两个数，并送入变量 `a`、`b` 中，然后调用 `max` 函数，把 `a`、`b` 的值传送给 `max` 函数的参数 `x`、`y`。在 `max` 函数中比较 `x`、`y` 的大小，把较大者通过中间变量 `m` 返回 (`return m`) 给主函数的变量 `c`，最后在屏幕上输出 `c` 的值。