

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21 世纪普通高校计算机公共课程规划教材

C 语言程序设计教程

张建勋 纪 纲 主 编

陈 渝 刘春萌 副主编



清华大学出版社

21 世纪普通高校计算机公共课程规划教材

TP312/2686

2008

C语言程序设计教程

张建勋 纪 纲 主 编
陈 渝 刘春萌 副主编

清华大学出版社
北京

内 容 简 介

本书注重教材的可读性和适用性,全书共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,基本输入、输出和顺序程序设计,选择结构程序设计,循环结构程序设计,数组,函数,指针,预处理命令,宏定义、复杂数据类型,文件等。

本书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“指点要津”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;附有大量的图表和程序,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。

本书可作为高等学校大学本科、高职高专学生“C语言程序设计”课程教学用书,也可作为全国计算机水平考试及各类短训班的培训教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

C语言程序设计教程/张建勋,纪纲主编. —北京:清华大学出版社,2008.2

(21世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-16813-3

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(2008)第005360号

责任编辑:梁颖 李玮琪

责任校对:徐俊伟

责任印制:李红英

出版发行:清华大学出版社 地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社总机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印刷者:北京市世界知识印刷厂

装订者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:21 字 数:504千字

版 次:2008年2月第1版 印 次:2008年2月第1次印刷

印 数:1~5000

定 价:29.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:028160-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高,以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。本规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评

审确定主题。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人：梁颖 liangying@tup.tsinghua.edu.cn

前 言

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，既具有高级语言的特点，又具有低级语言的许多特点，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，现在都可以用 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，许多高校的高级语言程序设计课程都主要以 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，更加强调学生的动手实践能力，因此，迫切需要一本内容新颖、体系合理、逻辑性强、通俗易懂的 C 语言教材。

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“指点要津”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。

全书由重庆工学院“C 语言程序设计”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 语言程序设计”教学的一线教师，有丰富的教学经验和软件开发能力。书中文字流畅、通俗易懂、概念清楚、深入浅出、例题丰富，实用性强。

第 1、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 章由纪纲教授编写，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。

本书可作为高等学校大学本科、高职高专学生“C 语言程序设计”课程教学用书，也可作为全国计算机水平考试及各类短训班的培训教材。

为了方便教学，我们开发了计算机网络考试系统，使用本书的院校可通过邮件“zhangjianxuncq@sina.com”向作者索取。

由于我们的水平有限，本书可能会有不尽如人意和疏漏之处，敬请读者批评指正，以便我们及时修改。

编 者

2007 年 11 月

目 录

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	2
1.1.1 C 语言的发展	2
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	8
1.4 运行 C 语言程序的基本步骤	14
小结	18
习题一	18
第 2 章 基本数据类型、运算符与表达式	21
2.1 C 语言的数据类型	22
2.2 常量、变量和标识符	23
2.3 简单数据类型与表示范围	27
2.3.1 整型数据	27
2.3.2 实型数据	31
2.3.3 字符型数据和字符串常量	33
2.3.4 各数据类型间的混合运算	37
2.3.5 简单数据类型的表示范围	38
2.3.6 数据的简单输出	39
2.4 C 语言的运算符与表达式	40
2.4.1 C 语言运算符简介	40
2.4.2 赋值运算符和赋值表达式	41
2.4.3 强制类型转换符	46
2.4.4 算术运算符和算术表达式	47
2.4.5 自增自减运算符	49
2.4.6 位运算符、位运算表达式	51
2.4.7 逗号运算符和逗号表达式	53
2.4.8 指针运算符、sizeof 运算符	55
2.5 运算符的优先级和结合性	55
小结	56
习题二	58

第 3 章 顺序结构程序设计	61
3.1 结构化程序设计	62
3.1.1 结构化程序设计概述	62
3.1.2 结构化程序设计的基本结构及其特点	63
3.2 C 语句概述	63
3.3 输入输出介绍	65
3.3.1 格式化输出函数 printf	65
3.3.2 格式化输入函数 scanf	68
3.3.3 字符输出函数 putchar	69
3.3.4 字符输入函数 getchar	70
3.4 顺序结构程序设计举例	71
小结	73
习题三	73
第 4 章 选择结构程序设计	75
4.1 关系运算符、逻辑运算符、条件运算符	75
4.1.1 关系运算符和关系表达式	75
4.1.2 逻辑运算符和逻辑表达式	76
4.1.3 条件运算符和条件表达式	77
4.2 选择结构的程序设计	78
4.2.1 if 语句	78
4.2.2 switch 语句	79
4.3 选择结构程序设计举例	81
小结	85
习题四	85
第 5 章 循环结构程序设计	87
5.1 while 语句	88
5.2 do-while 语句	90
5.3 for 语句	93
5.4 goto、break、continue 语句	95
5.5 循环的嵌套	97
5.6 程序举例	98
小结	101
习题五	101
第 6 章 数组	103
6.1 一维数组的定义和引用	105

6.1.1	一维数组的定义	105
6.1.2	一维数组元素的引用	106
6.1.3	一维数组的初始化	107
6.1.4	一维数组程序举例	108
6.2	二维数组	110
6.2.1	二维数组的定义	111
6.2.2	二维数组元素的引用	111
6.2.3	二维数组的初始化	112
6.2.4	二维数组程序举例	113
6.3	字符数组	116
6.3.1	字符数组的定义	116
6.3.2	字符数组的初始化	116
6.3.3	字符数组的引用	117
6.3.4	字符串	118
6.3.5	字符串的输入输出	119
6.3.6	字符串操作函数	122
6.3.7	字符数组应用举例	124
小结		127
习题六		128
第7章	函数	129
7.1	函数概述	132
7.1.1	函数与模块化程序设计方法	132
7.1.2	函数的分类	133
7.1.3	主函数	134
7.2	函数定义和调用	134
7.2.1	函数定义	134
7.2.2	函数调用	138
7.3	函数参数传递	140
7.3.1	传值调用	140
7.3.2	传址调用	141
7.4	函数与数组	143
7.5	函数与指针	148
7.5.1	返回指针的函数	149
*7.5.2	指向函数的指针	150
7.6	函数与结构	152
7.6.1	结构指针及结构变量的传址调用	152
7.6.2	结构型函数	154
7.6.3	结构指针型函数	155

7.7 递归函数.....	156
*7.8 命令行参数.....	161
7.9 标准库函数.....	163
7.10 程序举例.....	164
小结	169
习题七	169
第 8 章 指针	172
8.1 指针与指针变量的概念.....	173
8.1.1 指针的概念.....	173
8.1.2 指针变量.....	173
8.2 指针变量的定义和引用.....	176
8.2.1 指针变量的定义.....	176
8.2.2 指针变量的引用.....	177
8.3 指针和地址运算.....	179
8.4 指针与数组.....	180
8.4.1 指针变量与数组.....	180
8.4.2 指针变量在一维数组中的应用.....	183
8.4.3 指针变量在多维数组中的应用.....	185
8.5 指针与字符串.....	191
8.5.1 字符串操作的特点及字符指针变量的引入.....	191
8.5.2 指向字符串的指针变量.....	191
8.6 指针数组与多级指针.....	194
8.6.1 指针数组.....	194
8.6.2 指针数组的使用.....	195
8.6.3 多级指针.....	196
8.7 指针变量与函数.....	198
8.7.1 函数的操作方式与指针变量.....	198
8.7.2 指针型函数的定义与使用.....	199
8.7.3 函数指针的定义与使用.....	200
8.7.4 与指针有关的函数参数传递方式.....	202
8.7.5 带参数的 main 函数和命令行参数.....	207
8.8 指针与动态内存分配.....	208
8.8.1 动态存储的概念.....	208
8.8.2 C 语言的动态存储管理方式.....	209
小结	211
习题八	214
第 9 章 编译预处理.....	220
9.1 宏定义.....	222

9.1.1	不带参数的宏定义	222
9.1.2	带参数的宏定义	224
9.2	文件包含处理	228
9.3	条件编译	232
	小结	236
	习题九	238
第 10 章	复杂数据类型	241
10.1	复杂数据类型概述	242
10.2	结构体	243
10.2.1	结构体类型的定义	243
10.2.2	结构体变量的定义和引用	244
10.2.3	结构体变量的赋值	247
10.2.4	结构体数组	248
10.3	线性链表	251
10.3.1	固定内存分配与动态内存分配的概念	251
10.3.2	链表的概念	253
10.3.3	动态存储分配链表的基本操作	256
10.4	共用体	259
10.4.1	共用体类型的定义	259
10.4.2	共用体变量的定义和引用	259
10.4.3	共用体变量的赋值	261
10.5	枚举类型	262
10.5.1	枚举类型的定义	262
10.5.2	枚举变量的说明与引用	262
10.6	自定义类型	264
10.7	复杂数据类型应用综合举例	266
	小结	269
	习题十	270
第 11 章	文件	276
11.1	文件的基本概念	277
11.2	文件的类别	279
11.3	文件指针	280
11.4	文件操作概述	280
11.5	文件的打开、读/写和关闭	281
11.5.1	文件的打开 (fopen 函数)	281
11.5.2	文件的关闭 (fclose 函数)	283
11.5.3	文件的读/写	283

11.6 文件的定位读/写	290
11.7 文件应用综合举例	291
小结	295
习题十一	295
附录 A C 语言的字符集-ASCII 字符表	296
附录 B C 语言的关键字	297
附录 C 运算符的结合性	298
附录 D C 语言的库函数	299
附录 E C 语言常见错误	306
附录 F Turbo C 常见错误提示英汉	311
附录 G 重庆市计算机二级等级考试大纲	315
参考文献	319

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
人类语言 (也称自动语言)	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

1. 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,

并阅读简单的 C 程序。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 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 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源文件组成,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) 每一个说明及每一条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,一条语句可以占多行,一行也可以有多条语句。

3) 算法的表示方法

算法的表示方法可用流程图的画法。

指点要津

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. 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 ALGOL 60。ALGOL 60 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 C 语言（取 BCPL 的第二个字母），并用之改写了 UNIX (UNIX 第五版) 操作系统。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 语言版本不仅符合 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，关键字都是小写字母书写的。

(2) 运算符丰富。共有 34 种运算符。C 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。一行可以写多条语句，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,但可从这些例子中了解到组成一个 C 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 程序一般用小写字母书写。

(2) 每个 C 源程序必须要有且只能有一个 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 程序是由函数构成的,函数是 C 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) 每条语句的结尾,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) 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”为预编译命令也称为文件包含命令,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。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。

对此将在后续章节中进行详细介绍。

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);          /*按指定格式输出*/
}
```

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, 使用了三个库函数: 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`。

(2) 也可以用 “`//`” 符号标识注释内容, 这种注释只在一行末有效, 在执行语句的前面和中间不能使用这种注释。

(3) 在例题中的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。说明部分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。这是编译型高级程序设计语言的一个特点。说明部分是 C 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输入某一数, 如 9, 按下回车键 (enter), 接着在屏幕上给出计算结果。