

◎ 高等学校计算机基础教材



C/C++程序设计

吴国凤 宣善立 主编



高等教育出版社
Higher Education Press

高等学校计算机基础教材

C/C++程序设计

吴国凤 宣善立 主编

高等教育出版社

内容简介

本书通过大量案例介绍 C/C++ 语言程序设计的基本概念、方法和技巧。全书共分 12 章, 主要内容包括: C 语言概述、基本程序设计、程序控制结构、数组、函数、指针、结构体与共用体、位运算、文件、C++ 基础知识、面向对象程序设计及应用程序实例等。

全书内容全面, 结构精炼, 系统性强, 例题丰富。为了便于教学, 另配《C/C++ 程序设计实验指导与题解》一书。

本书适用于高等院校理工科各专业程序设计课程教材, 也可作为计算机等级考试的参考用书。

本书电子教案及案例源代码等相关教学资源均可以从高等教育出版社高等理工教学资源网上下载, 网址为 <http://www.hep-st.com.cn>, 也可以与作者 (wgf2867@126.com) 联系, 获取更多教学资源。

图书在版编目 (CIP) 数据

C/C++ 程序设计/吴国凤, 宣善立主编. —北京: 高等教育出版社, 2006.8

ISBN 7-04-018906-2

I. C... II. ①吴...②宣... 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 (2006) 第 076788 号

策划编辑	雷顺加	责任编辑	萧 潇	封面设计	于文燕	责任绘图	黄建英
版式设计	陆瑞红	责任校对	胡晓琪	责任印制	毛斯璐		

出版发行 高等教育出版社

社 址 北京市西城区德外大街 4 号

邮政编码 100011

总 机 010-58581000

经 销 蓝色畅想图书发行有限公司

印 刷 北京市联华印刷厂

开 本 787 × 1092 1/16

印 张 20.75

字 数 500 000

购书热线 010-58581118

免费咨询 800-810-0598

网 址 <http://www.hep.edu.cn>

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

网上订购 <http://www.landaco.com>

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

畅想教育 <http://www.widedu.com>

版 次 2006 年 8 月第 1 版

印 次 2006 年 8 月第 1 次印刷

定 价 26.00 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 18906-00

前言

本书是根据教育部最新提出的《关于进一步加强高等学校计算机基础教学的意见》中有关“计算机程序设计基础”课程的教学要求而编写的。

为了培养高素质人才,各高等院校均已将计算机基础教育放到了很重要的位置,计算机基础教育不仅包括大学计算机基础,还包括计算机程序设计基础。通过程序设计语言的学习,学习者可以掌握程序设计的方法与技术,培养程序设计的能力。为了满足程序设计基础教学改革的需要,我们组织了有关高校长期在教学第一线工作的教师编写了《C/C++程序设计》一书,供广大高校的师生们选用。为适应教学方式的变革,除了编写主教材之外,还编写了配套的辅助教材、课件、电子教案及相应的程序设计素材。

本书共12章,主要内容包括:C语言概述、基本程序设计、程序控制结构、数组、函数、指针、结构体与共用体、位运算、文件、C++基础知识、面向对象程序设计及应用程序实例等。本书全面、系统地讲述了C/C++语言的基本概念、编写程序的方法以及程序设计的技巧。本书内容丰富,概念清晰,组织精炼,系统性强,同时突出了学生程序设计能力的培养。另外,每章还配备了大量的习题,便于学生练习掌握主要知识点。

本书由吴国凤、宣善立任主编,姚合生、黄晓梅任副主编。各章编写分工如下:第1、9章由吴国凤编写,第2章由谢文佩编写,第3、12章由宣善立编写,第4、8章由黄晓梅编写,第5章由王金玲编写,第6、7章由姚合生编写,第10、11章由红光编写,全书由吴国凤、宣善立统编定稿。参加本书编写的还有李明、偶春生、王卫、林杰华、娄彦山、方乃义、冷金麟、黎杰等。

在本书的编写过程中得到了合肥工业大学计算机与信息学院胡学钢、王浩教授和高等教育出版社领导和相关编辑的大力支持以及许多同行的支持和帮助,在此一并表示最真诚的谢意。

由于编者水平有限,书中难免有疏忽、错误之处,恳请专家和读者批评指正。

编 者

2006.6

郑 重 声 明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010) 58581897/58581896/58581879

传 真：(010) 82086060

E - mail: dd@hep.com.cn

通信地址：北京市西城区德外大街 4 号

高等教育出版社打击盗版办公室

邮 编：100011

购书请拨打电话：(010)58581118

目 录

第1章 C语言概述 1

1.1 C语言简介 1

1.1.1 C语言的由来 1

1.1.2 C语言的特点 2

1.2 C语言程序的基本结构 4

1.2.1 C程序实例 4

1.2.2 C程序结构特点 6

1.3 C语言的语法 7

1.3.1 字符集 7

1.3.2 关键字 9

1.3.3 标识符 10

1.3.4 分隔符 10

1.4 程序设计与算法 11

1.4.1 计算机算法 11

1.4.2 程序设计基础知识 13

1.5 典型例题精解 16

本章小结 17

习题一 17

第2章 基本程序设计 19

2.1 数据类型 19

2.2 常量及其类型 20

2.2.1 整型常量的表示方法 20

2.2.2 实型常量的表示方法 21

2.2.3 字符常量 21

2.2.4 字符串常量 22

2.2.5 符号常量 22

2.3 变量及其类型 23

2.3.1 变量及其说明 23

2.3.2 整型变量 23

2.3.3 实型变量 26

2.3.4 字符变量 27

2.4 运算符和表达式 28

2.4.1 算术运算符和算术表达式 28

2.4.2 赋值运算符和赋值表达式 30

2.4.3 变量赋值及表达式计算时的 数据类型转换 31

2.4.4 逗号运算符和逗号表达式 33

2.4.5 条件运算符和条件表达式 33

2.4.6 sizeof 长度运算符 34

2.5 数据的输入/输出 34

2.5.1 数据输入/输出概念 34

2.5.2 字符数据的输入/输出 35

2.5.3 格式输入与输出 36

2.6 典型例题精解 41

本章小结 43

习题二 43

第3章 程序控制结构 47

3.1 顺序结构 47

3.1.1 C语言中的语句 47

3.1.2 顺序结构程序设计 48

3.2 关系运算与逻辑运算 50

3.2.1 关系运算符和关系表达式 50

3.2.2 逻辑运算符和逻辑表达式 51

3.2.3 运算的优先级 52

3.2.4 构造条件 53

3.3 分支结构 54

3.3.1 由 if (else) 语句构成的 三种分支 54

3.3.2 switch 语句 57

3.4 循环结构 59

II 目录

3.4.1 while 语句	59	5.2.2 地址传递方式	120
3.4.2 do-while 语句	60	5.3 标准库函数	121
3.4.3 for 语句	61	5.3.1 库函数的调用	122
3.4.4 转移语句	63	5.3.2 库函数的分类	122
3.4.5 循环的嵌套	64	5.3.3 常用库函数	122
3.5 典型例题精解	65	5.4 数组与函数	125
本章小结	72	5.4.1 函数参数为数组元素	125
习题三	73	5.4.2 函数参数为数组名	126
第4章 数组	80	5.5 函数的嵌套调用与递归调用	128
4.1 数组的概念	80	5.5.1 函数的嵌套调用	128
4.2 一维数组	81	5.5.2 函数的递归调用	129
4.2.1 一维数组类型说明	81	5.6 变量的作用域和存储类型	131
4.2.2 一维数组元素的表示方法	81	5.6.1 局部变量和全局变量	131
4.2.3 一维数组的初始化	82	5.6.2 变量的存储类型	133
4.2.4 一维数组的应用	84	5.6.3 内部函数与外部函数	138
4.3 多维数组	86	5.7 编译预处理	139
4.3.1 多维数组的概念	86	5.7.1 宏定义	139
4.3.2 二维数组的说明	87	5.7.2 文件包含	142
4.3.3 二维数组元素的存储顺序	88	5.7.3 条件编译	144
4.3.4 二维数组的初始化	88	5.8 典型例题精解	145
4.3.5 二维数组的应用	90	本章小结	150
4.4 字符数组和字符串	92	习题五	151
4.4.1 字符数组的说明	92	第6章 指针	157
4.4.2 字符数组的初始化	92	6.1 指针的基本概念	157
4.4.3 字符串的输入/输出	93	6.1.1 内存、地址和指针	157
4.4.4 字符串处理函数	94	6.1.2 指针变量的定义与引用	158
4.5 典型例题精解	99	6.1.3 指针变量作为函数参数	161
本章小结	104	6.1.4 多级指针的概念	163
习题四	104	6.1.5 指针的运算	164
第5章 函数	110	6.2 指针与数组	165
5.1 概述	110	6.2.1 指针与一维数组	165
5.1.1 函数的概念	110	6.2.2 指针与二维数组	168
5.1.2 函数的定义	111	6.2.3 指针与字符串	172
5.1.3 函数的调用	113	6.2.4 指针数组	175
5.2 函数间的数据传递	118	6.3 指针与函数	179
5.2.1 值传递方式	118	6.3.1 函数指针	179
		6.3.2 返回指针的函数	181

6.3.3 带参数的主函数	182
6.4 典型例题精解	183
本章小结	186
习题六	188

第7章 结构体与共用体

7.1 结构体	195
7.1.1 结构体类型的定义	195
7.1.2 结构体变量的说明	196
7.1.3 结构体变量的初始化	198
7.1.4 结构体变量的引用	199
7.1.5 结构体数组	200
7.1.6 结构体指针	203
7.1.7 结构体与函数	205
7.1.8 动态内存分配与链表	208
7.2 共用体	216
7.2.1 共用体类型的定义	216
7.2.2 共用体变量的说明	216
7.2.3 共用体变量的引用	217
7.3 枚举类型	220
7.3.1 枚举类型的定义	220
7.3.2 枚举变量的说明及引用	220
7.3.3 枚举类型的应用	222
7.4 用户定义类型	222
7.5 典型例题精解	224
本章小结	228
习题七	229

第8章 位运算

8.1 位运算的基本概念	237
8.2 计算机内的数据表示	237
8.3 位运算	238
8.3.1 逻辑位运算	238
8.3.2 移位运算	242
8.4 位域	244
8.4.1 位域的定义及位域变量的说明	244
8.4.2 位域变量的使用	245

8.5 典型例题精解	247
本章小结	249
习题八	249

第9章 文件

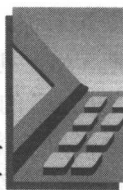
9.1 概述	252
9.1.1 文件的基本概念	252
9.1.2 文件的分类	253
9.2 文件类型指针	254
9.3 文件的打开与关闭	255
9.3.1 文件的打开	255
9.3.2 文件的关闭	256
9.4 文件的读/写	257
9.4.1 字符输入/输出函数	257
9.4.2 文件的字符串输入/输出函数	259
9.4.3 文件的格式化输入/输出函数	260
9.4.4 文件数据块的输入/输出函数	261
9.4.5 整数输入/输出函数	263
9.5 文件的定位操作	264
9.6 文件的错误检测	266
9.7 典型例题精解	267
本章小结	268
习题九	269

第10章 C++基础知识

10.1 C++程序结构	272
10.1.1 一个简单的C++程序	272
10.1.2 C++程序的结构特点	273
10.2 C++与C语言的比较	274
10.2.1 输入和输出	274
10.2.2 注释方式	274
10.2.3 常量	275
10.2.4 变量与数据类型	275
10.2.5 简单的输入/输出格式控制	276
10.2.6 动态内存分配	277

IV 目录

10.2.7 函数.....	278	本章小结.....	300
10.3 引用.....	278	习题十一.....	300
10.4 内联函数.....	279		
10.5 典型例题精解.....	280	第 12 章 应用程序实例.....	303
本章小结.....	281	12.1 程序设计的过程和方法.....	303
习题十.....	282	12.1.1 问题定义.....	303
第 11 章 面向对象程序设计.....	286	12.1.2 设计方法.....	304
11.1 类与对象.....	286	12.1.3 系统模块分解.....	305
11.1.1 面向对象的基本概念.....	286	12.1.4 模块设计.....	305
11.1.2 类的定义及实现.....	287	12.1.5 编码.....	307
11.1.3 对象的定义及实现.....	289	12.2 学生成绩管理系统源代码.....	308
11.2 构造函数与析构函数.....	290	本章小结.....	315
11.3 继承与派生.....	291		
11.3.1 概述.....	291	附录 1 常用字符与 ASCII	
11.3.2 单一继承.....	292	代码对照表.....	316
11.3.3 多重继承.....	293		
11.4 多态性.....	295	附录 2 运算符的优先级和	
11.4.1 多态性简介.....	295	结合性总表.....	318
11.4.2 函数重载.....	296		
11.5 典型例题精解.....	297	参考文献.....	320



C 语言概述

【内容概述】

通过本章的学习,了解 C 语言的发展简史、特点、与其他高级语言相比的不同点;掌握 C 语言基本程序结构以及书写格式;掌握 C 语言中定义标识符的方法;掌握 ANSI C 中使用的关键字;了解程序设计的基本知识和计算机算法的应用。

【学习目标】

- 了解 C 语言的特点。
- 掌握 C 程序的构成、main() 函数及其他函数的结构。
- 掌握 C 程序的书写格式。
- 掌握关键字、标识符、C 语言的语句。
- 了解程序设计的基本知识和计算机算法的应用。
- 掌握标题文件(头文件)的使用。

1.1 C 语言简介

1.1.1 C 语言的由来

C 语言是目前世界上最流行和实用的一种计算机高级程序设计语言。它适合于作为系统描述语言,既可以用来编写系统软件,也可以用来编写应用软件。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的,后随其宿主系统 UNIX 的普及而引起注意,又因其本身的高效、灵活、功能丰富、表达力强、移植性好等特殊优点,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。

C 语言的原型是 ALGOL60 语言。1963 年,剑桥大学将 ALGOL60 语言发展成为 CPL (Combined Programming Language, 复合程序设计语言)。CPL 语言接近硬件一些,但规模较大,难以实现。1967 年剑桥大学的马丁·理查德对 CPL 进行了简化,推出了 BCPL (基本复合程序设计语言)。1970 年美国贝尔实验室的肯·汤普逊对 BCPL 进行了进一步简化,突出硬件处理能力,并取“BCPL”的第一个字母“B”作为新语言的名称,并且用 B 语言编写了第一个 UNIX



操作系统。但 B 语言过于简单, 功能有限, 在 1972 年, 贝尔实验室的布朗·W·卡尼汉和丹尼斯·M·利奇对 B 语言进行了完善和扩充, 在 B 语言的基础上最终设计出一种新的语言, 他取了“BCPL”的第二个字母“C”作为这种语言的名字。这就是大名鼎鼎的 C 语言。C 语言保持了 BCPL 语言和 B 语言精炼、接近硬件的优点, 又克服了它们过于简单、数据无类型、功能有限的缺点。后来经过多次改进, 在贝尔实验室内部使用。1978 年以后, C 语言被移植到各种机型上。由于它作为系统描述语言既可以用来编写系统软件, 又可以用来编写应用软件, 所以很快受到欢迎。许多程序员纷纷从别的语言转到 C 语言编程, 从而使 C 语言在当时的软件开发中几乎一统天下。

1977 年, 为了让 C 语言脱离 UNIX 操作系统, 成为在任何计算机上都能运行的通用计算机语言, 布朗·W·卡尼汉和丹尼斯·M·利奇(K&R)合著了著名的《C 程序设计语言》一书, 对 C 语言的语法进行了规范化的描述, 成为当时的标准, 通常简称为《K&R》标准。但是, 在《K&R》中并没有定义一个完整的标准 C 语言, 后来由 ANSI (American National Standards Institute, 美国国家标准协会) 在此基础上制订了一个 C 语言标准, 于 1983 年发表, 通常称为 ANSI C 或标准 C。

现在使用的各种 C 语言编译系统虽然基本部分是相同的, 但也有一些不同。在微型机上使用的有 Microsoft C、Turbo C、Quick C、Borland C 等, 它们的不同版本又略有差异。因此, 在使用一个系统之前, 要了解所用的计算机系统配置的 C 编译系统的特点和规定。

在 C 的基础上, 1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言, 成为一种面向对象的程序设计语言。C++ 目前流行的最新版本是 Borland C++、Symantec C++ 和 Microsoft Visual C++。

C++ 提出了一些更为深入的概念, 它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间, 为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法, 因而也增加了整个语言的复杂性, 掌握起来有一定难度。但是, C 是 C++ 的基础, C++ 语言和 C 语言在很多方面是兼容的。因此, 掌握了 C 语言, 就为进一步学习 C++ 打下了基础, 能以一种熟悉的语法来学习面向对象的语言, 从而达到事半功倍的目的。

1.1.2 C 语言的特点

C 语言同时具有汇编语言和高级语言的优势, 它可以对内存地址进行读/写, 可直接对硬件、二进制位进行操作。它把高级语言面向过程和低级语言与硬件关系密切的优点有机地结合起来, 广泛应用于系统软件、应用软件、数据处理以及科学计算等领域。C 语言之所以成为目前世界上使用最广泛的程序设计语言, 并被作为适应近代软件工程需要的面向对象语言 C++ 的基础语言, 是由于 C 语言的诸多突出优点所决定的。

1. 语言简洁紧凑, 使用方便

ANSI C 一共只有 32 个关键字和 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 (与 Pascal 语言的比较), 如表 1-1 所示。

表 1-1 C 语言与 Pascal 语言比较

C 语言	Pascal 语言	含 义
{ }	BEGIN...END	复合语句
if(e) s;	IF(e) THEN s	条件语句



续表

C 语言	Pascal 语言	含 义
int i;	VAR i:INTEGER	定义 i 为整型变量
int a[10];	VAR a:ARRAY[1..10] OF INTEGER	定义整型数组 a
int f();	FUNCTION f():INTEGER	定义 f 为返回值整型的函数
int *p;	VAR p:↑INTEGER	定义 p 为指向整型变量的指针
i+=2;	i=i+2	赋值 i+2⇒i
i++,++i;	i=i+1	i 自增值

结论: C 语言比 Pascal 语言简练, 源程序短。

2. 运算符丰富

C 语言共有 44 种运算符 (见附录 2)。C 把括号、赋值、强制类型转换等都作为运算符处理, 从而使得 C 的运算符极其丰富, 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。例如:

C 语言具有复合的赋值运算符+、-、*、/、%、= (加、减、乘、除) 等, >>=、<<= (右移、左移) 等, &、^、|、= (与、或、非) 等。如 $x+=5$ 等价于 $x=x+5$ 。

C 语言有条件运算符?:可代替简单的 if-else 语句。若表示“如果 x 小于或等于 0, y 为 0; 否则 y 为 1”, 可以表示为“ $y=x \leq 0 ? 0 : 1$ ”。C 语言中连赋值这种操作都定义为运算符, 也就是说赋值操作本身可以作为表达式的一部分, 参与运算。如: “if((p=malloc(sizeof(int)))!=NULL) {printf("Error!");exit(1);}”。

3. 数据结构类型丰富

C 语言具有整型、实型、字符型、数组、指针、结构体、共同体等数据类型, 能方便地构造更加复杂的数据结构 (如使用指针构造链表、树、栈), 为处理各种复杂数据类型提供了实用的手段。

4. 模块化结构

C 语言是一种模块化的程序设计语言。C 语言的程序是由一系列函数构成的, 这些函数可以自己设计, 也可以由标准函数提供。C 语言程序是由最顶层的函数 main()按照一定的组织层次调用标准函数或自己设计的函数而形成的。

5. 具有结构化的控制语句

C 语言具有多种结构化的控制语句, 提供了 if-else 语句、while 语句、do-while 语句、switch 语句、for 语句, 可以很容易地实现结构化的各种基本结构, 用来设计结构化程序。

6. 允许直接访问物理地址, 进行位操作

C 语言能实现汇编语言的大部分功能, 可以对硬件进行操作。

① C 语言可以直接操作计算机硬件, 如寄存器、各种外设 I/O 端口等。

② C 语言的指针可以直接访问内存物理地址。

③ C 语言类似汇编语言的位操作可以方便地检查系统硬件的状态。

因此, C 语言具有高级语言的功能, 又具有低级语言的许多功能, 可以用来编写系统软件。

C 语言的这种双重性, 使得它既可以作为系统描述语言, 又可以作为程序设计语言。因此, C



语言被称为高级语言中的低级语言。

7. C语言对语法限制不严格, 程序设计灵活

C语言不检查数组下标越界, 不限制对各种数据转化(编译系统可能对不恰当的转化进行警告, 但不限制), 不限制指针的使用, 程序正确性由程序员保证。

实践中, C语言程序编译时会提示“警告错误”或“严重错误”。“警告错误”表示用户使用的语法可能有问题, 但是有时可以忽略, 程序仍然可以完成编译工作, 然后运行(但是一般情况下“警告错误”往往意味着程序真的有问题, 应该认真地检查)。“严重错误”是不能忽略的, 编译系统发现严重错误, 就不会产生目标代码。

8. 生成的目标代码质量高, 程序执行效率高

C语言的一些操作是直接对应于实际计算机硬件的, 在很多方面具有汇编语言的特性, 因此能充分反映计算机硬件的功能, 所生成的代码效率极高。有资料显示C语言只比汇编代码效率低10%~20%。

9. 可移植性好

由于C语言的标准化以及C程序的输入/输出、内存管理等操作采用C库函数实现, 不仅使得C编译程序很容易在不同的机器系统上实现, 而且使得用户C程序可以不做修改或只做少量修改就能在不同的机器系统上运行。

C语言既具有高级语言的特点, 又具有低级语言的特点, 集二者的优点于一身, 数据结构丰富, 具有结构化的控制语句, 在使用C编写程序的时候限制少、灵活性大、功能强, 可以编写任何类型的程序。通过学习C语言, 可以掌握基本的数据结构和类型, 掌握结构化程序设计的特点, 掌握编写程序基本的逻辑知识, 为今后的程序设计以及软件的使用打下良好的基础。

1.2 C语言程序的基本结构

C语言的程序是由基本元素(包括常量、变量、函数、表达式以及C语言规定的关键字、数据类型和一定的语法规则)构成的。下面通过几个简单的C语言程序, 分析C语言程序的结构、特点和相关的规则, 以使读者对C语言有个初步感性认识, 从中了解C语言程序设计结构化程序设计的思想。

1.2.1 C程序实例

【例 1.1】 输出字符串。

```
#include <stdio.h>
void main(void)
{
    printf("welcome!\n");
}
```

程序运行结果如图 1-1 所示。

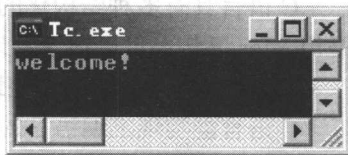


图 1-1 程序运行结果

这是一个最简单的C语言程序, 它把字符串“welcome!”显示在屏幕上。在程序中, `main()` 称做“主函数”, 在任何一个C程序中都必须有且只有一个 `main()` 函数。在 `main()` 函数中有一个或多个C语句, 用一对大括号 `{ }` 括起来, 大括号仅成对出现。在本例的程序中, 只有一



条 C 语句。这条 C 语句是一个输出函数，作用是向屏幕输出字符串信息，它将双引号之间的内容原样输出到屏幕。C 语言中没有输入/输出语句，输入/输出是通过 C 本身提供的标准函数来实现的，它的原型放在<stdio.h>头文件中。“\n”是换行符，它的作用是将光标移到下一行的开始处。

【例 1.2】 求 a 的平方根。

```
#include <stdio.h>
#include <math.h>
void main(void)                                /*求 a 的平方根*/
{
    float a=9, s_root;
    s_root =sqrt(a);                            /*求平方根*/
    printf("s_root=%f\n",s_root);              /*输出结果*/
}
```

程序运行结果如图 1-2 所示。



图 1-2 程序运行结果

程序中，/*...*/表示注释，注释部分不参与也不影响程序的运行，只是用来帮助人们阅读和理解程序的，可以加在程序的任何部分。程序中的 float 是 C 语言的关键字，用来说明变量为单精度实型；sqrt 是求平方根函数。C 语言规定，在使用数学函数时，一定要加上头文件：#include <math.h>。

printf()函数用于将结果输出到屏幕，“%f”为格式符，表示在这个位置上将有一个实型数值。

C 程序中的语句最重要的一个特点就是每条基本语句的后面都要有一个分号。

【例 1.3】 输入两个整数，输出两者中较大的数。

```
#include <stdio.h>
void main(void)
{
    int x,y,z;
    scanf("%d,%d",&x,&y);                      /*任意输入两个数*/
    z=imax(x,y);
    printf("\nThe Max number is: %d\n",z);
}

int imax(int x,int y)
{
    int z;
    if (x>y) z=x;
    else z=y;
    return(z);
}
```

若输入数据为：4,6↵

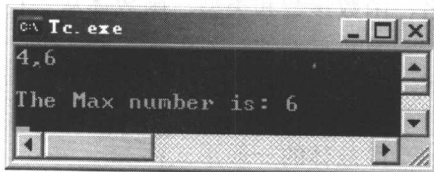


图 1-3 程序运行结果

程序运行结果如图 1-3 所示。

此程序包括两个函数，一个是主函数 main()，另一个是子函数 imax()，用于求两个数 x、y 中较大的数。在执行时，先由 scanf() 函数从键盘读取两个数据，这时由用户从键盘输入 4, 6↵ (↵表示回车键)。此时 x 被赋值 4，y 被赋值 6。然后执行函数调用语句：z=imax(x,y)，将 x、y 的值传入 imax 函数中。在 imax 函数中经过判断后，z 的值就是两个数的较大值。用 return 语



句将 *z* 的值返回函数调用处，然后继续执行主函数，直到程序结束。

1.2.2 C 程序结构特点

从上面三个例子可以看出 C 语言程序结构主要由注释、编译预处理以及程序主体三部分组成。

1. 注释

C 程序中可以用 `/*...*/` 对任何部分进行注释，一个好的程序都应有必要的注释以提高程序的可读性。在编写程序过程中，要养成使用注释的良好习惯，这对软件的维护相当重要。`/*` 和 `*/` 必须成对使用，且 `/*` 和 `*/` 之间不能有空格，否则会出错。

注释一般分为两种：一种为序言性注释，用于程序的开头，说明程序或模块的名称、用途、编写时间、使用变量等；另一种为注解性注释，用于程序的语句中，对难懂的部分加以说明。

注释可以出现在程序的任何位置，可以独占一行或多行。程序编译时，注释不参与编译，也不会出现在目标程序中。

2. 编译预处理

每个以符号 `#` 开头的行，称为编译预处理行。为了方便用户使用，C 编译系统提供了大量的内部函数，标准 C 中提供了 100 多条库函数，Turbo C 和 MS C 中提供了 300 多条库函数。这些函数都放在函数库 (`.lib`) 中，而把函数的说明放在标题文件 (也称头文件) 中，通过 `#include` 命令来完成。其一般形式为：

`#include <文件名> 或 #include "文件名"`

在 C 语言程序中，定义一条 `#include` 语句，就相当于将 `#include` 语句后面的文件的全部内容写在该位置上。将被包含文件嵌入到源文件，由编译预处理完成，即在编译之前完成。

利用 C 程序的函数式结构可以非常轻松地编写一些功能强大的程序。使得 C 程序非常容易实现模块化，便于阅读和维护。

3. 程序主体

① C 程序是由一个或多个函数构成的。每个 C 程序有且仅有一个主函数，函数名规定为 `main()` 函数。除主函数外，可以有一个或多个其他函数，并且在主函数中进行调用。

② 每个函数 (包括主函数) 的定义分为两个部分：函数头部和函数体。函数头部指定函数名、函数参数和类型；函数体从 `{` 开始，到 `}` 结束，函数内有语句。

函数头部的一般形式：

类型 函数名(形参 1, 形参 2, ...)

形式参数的说明

函数体的一般形式：

```
{  
    说明性语句  
    执行语句  
}
```

说明性语句主要对函数体内的变量说明其类型。执行部分可以是 C 语言语句，也可能是库函数或自己设计的函数调用语句。



③ C 程序总是从 `main()` 函数开始执行, 最终在 `main()` 函数中结束。`main()` 函数可以位于程序的任意位置。

④ C 语言的变量在使用之前必须先定义其数据类型, 未经定义的变量不能使用。定义变量类型的语句应在可执行语句的前面, 如例 1.3 中 `main()` 函数中的第一个语句就是变量定义语句, 它必须放在第一个执行语句 `scanf()` 的前面。

⑤ C 程序中的语句最后总要有个分号 (;) 作为每条语句的结束。但要注意, 函数说明语句后面无分号。如例 1.3 中 `void main(void)` 和 `int imax(int x, int y)` 后面不能加分号。

⑥ C 语言没有行的概念, 程序的书写格式比较自由, 可以在一行中写若干条语句, 也可以将一条语句写在多行上。习惯采用小写字母、缩进格式。



重点

① C 程序的基本结构是函数, 函数是完成某个功能的基本单位。

② C 函数从左花括号开始, 到对应的右花括号结束。

③ `main()` 可以位于程序的任何位置, 但 C 程序执行时, 总是从 `main()` 函数开始, 最终在 `main()` 函数中结束。



技巧

为避免遗漏, 对使用的符号实现配对, 例如注释符号、函数体的起止标识符 (花括号)、圆括号等。在输入时, 可连续输入这些起止标识符, 然后再在其中插入语句来完成内容的编辑。在起止标识符嵌套以及相距较远时, 这样做更有必要。

1.3 C 语言的语法

正如人类的自然语言具有其语法规则一样, C 语言也规定了它的语法。为了按照一定的语法规则构成 C 语言的各种成分 (如变量、表达式、语句等), C 语言规定了基本语法单位。

基本语法单位是具有独立语法意义的最小语法成分, 在 C 语言中基本语法单位被称为单词, 单词包括保留字、标识符、常量、运算符、分隔符、注释符 6 种。组成单词的基本符号是字符, 标准 C 及大多数 C 程序使用的字符集是 ASCII 字符集。

本节主要介绍字符集、保留字、标识符, 其他内容将在后续章节介绍。

1.3.1 字符集

字符是 C 语言最基本的元素, C 语言的字符集是书写源程序清单时允许出现的所有字符的集合, 也就是说, 在 C 语言的源程序清单中不能出现字符集以外的字符。字符集主要由字母、数字、键盘符号及转义符号构成。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)。

① 英文字母: A~Z, a~z (52 个)。

② 数字: 0~9 (10 个)。

③ 键盘符号: 33 个, 如表 1-2 所示。



表 1-2 键盘符号表

符 号	含 义	符 号	含 义	符 号	含 义
+	加号	[	左方括号	^	异或号
-	减号	]	右方括号	@	at 圈号
*	星号	{	左花括号	;	分号
/	正斜杠	}	右花括号	:	冒号
<	小于号	=	等号	'	单引号
>	大于号	~	波浪号	"	双引号
_	下划线	!	惊叹号		或号
\	反斜杠	#	井号	&	与号
(	左括号	%	百分号	?	问号
)	右括号	.	小数点	\$	美元符号
`	重音号	,	逗号	_	空格

④ 空白符：空白符是空格、制表符（跳格）、换行符（空行）的总称。空白符除了在字符、字符串中有意义外，其他位置的空白符均被编译系统忽略。空白符在程序中只是起到间隔作用。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。

⑤ 转义字符。

转义字符是由反斜杠（\）开头，后面跟一个字符或数字组成，用来表示一个特定含义的字符。例如：“\n”用来表示回车换行符。常用的转义字符如表 1-3 所示。

表 1-3 转义字符表

转义字符	含 义	转义字符	含 义
\n	回车换行符	\0	空字符（NULL）
\v	垂直制表符	\'	单引号
\r	回车符	\"	双引号
\a	响铃警报	\\	反斜杠字符
\t	水平制表符（Tab 键）	\ddd	1~3 位八进制数 ddd 对应的符号
\b	退格字符（向左删除一个字符）	\xhh	1~2 位十六进制数 hh 对应的符号
\f	换页符		

转义字符是一种特殊形式的字符常量，表示将转义符“\”后的字符原来的含义进行转换，变成某种另外特殊约定的含义。