

高等学校计算机基础教育规划教材

C语言程序设计 实用教程

鲍广华 钦明皖 主编



清华大学出版社

高等学校计算机基础教育规划教材

C语言程序设计 实用教程

鲍广华 钦明皖 主 编
王 虎 胡 勇 副主编

清华大学出版社
北京

内 容 简 介

本书作为普通高等院校计算机基础教学的教材,内容丰富,详略得当,深入浅出,通俗易懂,便于自学。全书共分12章,内容包括C语言概述、基本数据类型、数据的输入输出、程序的控制结构、数组、函数、指针、结构体、预编译、位运算、文件及综合实例。

本书可作为普通高校非计算机专业的教材或成教、职教计算机专业的教材,也可作为参加计算机等级考试人员及其他计算机自学者的自学教材。

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 13701121933

图书在版编目(CIP)数据

C语言程序设计实用教程/鲍广华,钦明皖主编. —北京:清华大学出版社,2015

高等学校计算机基础教育规划教材

ISBN 978-7-302-40502-3

I. ①C… II. ①鲍… ②钦… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第136990号

责任编辑:袁勤勇

封面设计:常雪影

责任校对:焦丽丽

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

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:185mm×260mm

印 张:23.75

字 数:552千字

版 次:2015年8月第1版

印 次:2015年8月第1次印刷

印 数:1~2000

定 价:38.00元

产品编号:041376-01

前言

“程序设计”是理工科学生的一门十分重要的基础课程,而 C 语言是一种国内外广泛使用的计算机语言。用 C 语言进行程序设计已成为计算机工作者的一个基本功,很多高校将“C 语言程序设计”(或类似名字)作为理工科学生的必修课。但是,由于 C 语言概念规则繁多,使用灵活易出错,很多学生在学习这门课时,感到吃力,虽然付出较多努力,但结果并不理想。

在我校,“程序设计基础 C”这门课程,也是非计算机专业理工科本科生的必修课。相对其他课程,该门课程的通过率也偏低。一直以来,我们不断研究,不断探索,不断改革,其目的,就是为了能使同学们学好这门课,练好这个基本功。

如何才能达到我们的目的,这其中要做的工作很多,而一本合适的教材对学生们来说,必不可少。目前关于 C 语言程序设计的教材有很多,但如何编写出一本让学生们学起来容易、看起来想看的教材,依然是我们日思夜想的目标,尽管我们编写的这本教材很可能没有完全实现我们的初衷,但至少我们的目标如此。

好的教材源于不懈的教学改革与教学实践。我们的教学改革,得到了学校相关部门一直以来的大力支持。教研项目“计算机程序设计基础课程教学模式改革与创新人才培养”于 2009 年获得学校批准立项(2012 年已结项),本次教材编写也是“安徽大学教学改革与建设项目”之一,本书为作者鲍广华承担的安徽大学教学改革与建设项目“C 语言程序设计实用教程”(项目类别:规划教材,项目编号:xjghjc1409)的研究成果。本教材的作者全部来自教学第一线,长期从事“程序设计基础 C”的教学与实践工作,潜心研究新形势下的教学对象、教学形式、教学内容、教学方法,积累了很多宝贵的经验。无论是在教研项目立项前、在研中还是结项后,我们的教学改革始终没有停歇,如何有效提高教学质量,如何通过“程序设计基础 C”这门课程的学习,使学生观察问题、分析问题及动手解决问题的能力得以全面提高,是我们一直以来追求的目标,也是我们编写本教材过程中始终牢记的。

全书以程序设计为主线,注重程序设计的思想和方法。通过具体问题或案例引出知识点,知识点介绍言简意明,通俗易懂,重点突出。联系实际编写的案例,与学生的生活、学习紧密相联,易于理解接受。案例中的算法分析,对编程思路给出分析,力求培养学生分析问题、解决问题的能力。注释形式的问题提出及分析,不仅有利学生更好理解程序代码,更是引导学生主动思维,以培养探索性、研究性学习习惯。对重点及必须掌握的难点内容,比如函数和指针这两部分内容,采用系列例题代码方式代之以传统单一例题方式,

针对学生难理解易搞错的问题,抓住问题的根本,详细讲解,举一反三,而对语法中的细枝末节则简要介绍或以实例等方式灵活处理,以期学生能够掌握真谛,从根本上真正掌握重点与难点,顺利学好本门课程,并希望通过本门课程的学习,促进学生综合能力的提高。

本书由鲍广华、钦明皖主编,王虎、胡勇副主编。第1章由郑珺露、张磊编写,第2章、第6章由王文兵编写,第3章、第7章由鲍广华编写,第4章、第5章由钦明皖编写,第8章、第9章由胡勇编写,第10章由郑珺露编写,第11章、第12章由王虎编写,附录由张磊编写。全书统稿由鲍广华完成,全书统稿后的通读与修改由鲍广华、钦明皖、王虎、胡勇完成。

参考的资料有的来自网络,有时很难记录到作者,所以有的没能标明出处,恳请涉及的作者原谅。如果引用了贵文,请来信说明,若再版我们将及时注明。

由于编著者水平有限,书中错误在所难免,恳请读者专家批评指正,我们将不胜感激。

最后,衷心感谢学校相关部门与教学部领导的大力支持,衷心感谢全体同仁的通力合作与鼎力相助,衷心感谢清华大学出版社广大员工的辛苦付出,衷心感谢亲人们的理解与支持,衷心感谢所有关爱帮助支持我的人,谢谢,谢谢你们!

鲍广华

2015年4月

目录

第 1 章 概述	1
1.1 C 语言简介	1
1.1.1 为什么要学习 C 语言	1
1.1.2 C 语言的诞生	3
1.1.3 C 语言的特点	4
1.1.4 C 语言源程序的结构	4
1.1.5 C 源程序的运行	6
1.2 程序与算法	7
1.2.1 程序与程序设计	8
1.2.2 算法	9
1.2.3 算法设计的基本方法	11
本章小结	18
习题 1	19
第 2 章 基本数据类型	20
2.1 C 语言的数据类型	20
2.2 常量与变量	22
2.2.1 标识符	22
2.2.2 常量	23
2.2.3 变量	24
2.3 整型数据	25
2.3.1 整型常量	25
2.3.2 整型数据在内存中的表示	25
2.3.3 整型变量	27
2.4 实型数据	30
2.4.1 实型常量	30
2.4.2 实型变量	30
2.5 字符型数据	31

2.5.1	字符型数据的表示	31
2.5.2	字符常量	32
2.5.3	字符变量	33
2.5.4	字符串常量	34
2.6	运算符和表达式	35
2.6.1	算术运算符及表达式	36
2.6.2	关系运算符及表达式	40
2.6.3	逻辑运算符及表达式	40
2.6.4	赋值运算符和赋值表达式	41
2.6.5	其他运算符及表达式	43
2.6.6	混合类型数据的运算	45
	本章小结	46
	习题 2	49
第 3 章	数据的输入输出	52
3.1	概述	52
3.1.1	C 语言的语句	52
3.1.2	C 语言中数据输入输出的实现	54
3.2	数据的格式化输入与输出	54
3.2.1	数据的格式化输出	54
3.2.2	数据的格式化输入	62
3.3	字符型数据的输入与输出	70
3.3.1	字符串的输入与输出	70
3.3.2	单个字符的输入与输出	71
	本章小结	76
	习题 3	80
第 4 章	程序的控制结构	84
4.1	程序的三种基本结构	84
4.2	顺序结构	85
4.3	分支结构	88
4.3.1	if 语句	88
4.3.2	switch 语句	95
4.4	循环结构	100
4.4.1	while 语句	101
4.4.2	do-while 语句	102
4.4.3	for 语句	103
4.4.4	三种循环语句的比较	105

4.4.5 循环结构的嵌套·····	105
4.4.6 辅助控制语句·····	107
4.5 应用举例·····	109
本章小结·····	118
习题 4·····	124
第 5 章 构造数据类型——数组 ·····	128
5.1 数组的概念·····	128
5.2 一维数组·····	129
5.2.1 一维数组的定义与初始化·····	129
5.2.2 一维数组的引用·····	131
5.2.3 一维数组的应用·····	132
5.3 二维及多维数组·····	137
5.3.1 二维数组的定义与初始化·····	137
5.3.2 二维数组的引用·····	140
5.3.3 二维数组的应用·····	140
5.3.4 多维数组的理解·····	142
5.4 字符型数组和字符串处理·····	142
5.4.1 字符型数组的概念与初始化·····	142
5.4.2 字符串的概念与初始化·····	143
5.4.3 字符型数组的输入和输出·····	146
5.4.4 字符串处理·····	148
5.5 应用举例·····	151
本章小结·····	158
习题 5·····	161
第 6 章 函数 ·····	166
6.1 概述·····	166
6.1.1 函数概述·····	166
6.1.2 函数分类·····	167
6.2 函数的定义、调用及返回·····	169
6.2.1 函数的定义·····	169
6.2.2 函数的调用·····	172
6.2.3 函数的返回值·····	176
6.3 函数参数的传递·····	177
6.4 函数的嵌套与递归调用·····	180
6.4.1 函数的嵌套调用·····	180
6.4.2 函数的递归调用·····	181

6.5	变量的作用域和存储类别	186
6.5.1	内部变量	186
6.5.2	外部变量	187
6.5.3	变量的存储类型	189
6.5.4	变量类别小结	191
6.6	应用举例	192
	本章小结	195
	习题 6	197
第 7 章	C 的指针	201
7.1	指针的概念、定义及基本操作	201
7.1.1	指针和指针变量	205
7.1.2	利用指针变量访问基本变量	209
7.2	用指针变量访问一维数组元素	216
7.2.1	指针变量的关系运算、算术运算	216
7.2.2	用指针变量访问一维数组元素	217
7.2.3	指针变量的基类型必须与所指变量的类型一致	219
7.3	指针变量做函数参数	220
7.4	用指针处理字符串	229
7.5	指针的其他应用	237
7.5.1	指针数组和数组指针	237
7.5.2	指针与函数	241
7.5.3	多级指针	243
7.6	main 函数的参数	244
7.7	应用举例	246
	本章小结	249
	习题 7	252
第 8 章	构造数据类型：结构、共用和枚举	262
8.1	结构体	262
8.1.1	概述	262
8.1.2	结构体数组	269
8.1.3	结构体指针变量	271
8.2	共用体	275
8.2.1	概述	275
8.2.2	应用举例	278
8.3	枚举	279
8.4	动态存储分配及链表	282

8.4.1 动态存储分配.....	282
8.4.2 链表.....	284
8.4.3 类型别名定义——typedef	290
本章小结.....	291
习题 8	291
第 9 章 预编译命令	295
9.1 概述	295
9.2 宏定义	296
9.2.1 无参数宏定义.....	296
9.2.2 带参数宏定义.....	297
9.2.3 宏定义的作用域.....	298
9.3 文件包含	298
9.3.1 概述.....	298
9.3.2 文件包含的作用.....	299
9.4 条件编译	299
9.4.1 第一种形式 #if 语句	300
9.4.2 第二种形式 #ifdef 语句	301
9.4.3 #ifndef 语句	301
本章小结.....	302
习题 9	302
第 10 章 位运算	305
10.1 位运算简介.....	305
10.2 移位运算.....	309
10.3 应用举例.....	311
本章小结.....	312
习题 10	312
第 11 章 数据的永久保存——文件	313
11.1 概述.....	313
11.1.1 文件的概念.....	313
11.1.2 文件指针.....	314
11.2 文件的打开与关闭.....	315
11.2.1 fopen 函数	316
11.2.2 fclose 函数	318
11.3 文件的读写.....	319
11.3.1 读写文件中字符的函数.....	319

11.3.2 读写文件中字符串的函数.....	322
11.3.3 格式化读写函数.....	324
11.3.4 数据块读写函数.....	325
11.4 文件的定位.....	328
11.5 文件的出错检测.....	330
11.6 文件操作实例.....	331
本章小结.....	333
习题 11	334
第 12 章 综合实例	336
12.1 概述.....	336
12.2 牛顿迭代法.....	336
12.3 穷举法求勾股数.....	337
12.4 回溯法求八皇后问题.....	338
12.5 一个简单的通讯录管理程序.....	340
本章小结.....	349
附录 A 运算符优先级和结合性.....	350
附录 B 常用字符 ASCII 码.....	352
附录 C C 程序集成开发环境——VC++ 6.0	354
附录 D C 常用标准库函数	365
参考文献.....	370

概 述

学习目标

了解为什么要学习 C 语言, C 语言有哪些特点, C 语言源程序的结构; 了解算法的概念及特征、程序及程序设计的概念、程序设计的一般步骤; 掌握算法的表示, C 语言源程序的运行。了解几种常见的典型算法。

重点、难点

重点: 算法、程序、程序设计的概念, 算法的表示。

难点: 用 N-S 图表示算法。

1.1 C 语言简介

1.1.1 为什么要学习 C 语言

C 语言是众多程序设计语言(又称计算机语言)中的一种。不同的程序设计语言, 各有各的特点, 而程序员们也是萝卜青菜各有所爱。但无论怎样, 没人能够否认, C 语言是一门伟大的语言, 一门承前启后的语言。C 语言如此旺盛的生命力, 如此之长的生存期, 就连 C 语言的发明人自己也没有料到。它的诞生是现代程序设计语言革命的起点, 是程序设计语言发展史中的一个里程碑。C 语言既有高级语言的特点, 同时又有汇编语言的特点, 它把高级语言的基本结构和语句与低级语言的实用性完美结合起来。

C 语言的数据类型、运算类型、表达式类型极其丰富, 不仅可以实现各种复杂数据结构的运算, 甚至可以实现现在其他高级语言中难以实现的运算。

C 语言之后, 以 C 为根基的 C++、C#、Java 以及 JavaScript、Objective-C 等程序设计语言层出不穷, 并在各自领域大获成功。但迄今为止, C 语言在系统编程、嵌入式编程等领域依然占据统治地位。这是因为 C 语言允许程序员直接操作硬件, 使得产生的目标代码质量高, 程序的执行效率高, 而嵌入式系统对程序运行的效率要求苛刻, 故嵌入式系统编程首选 C 语言。操作系统的编写, C 语言更是不二之选。若用 C 语言加上一些汇编语言子程序, 就更能显示出 C 语言的优势。

目前比较权威的编程语言流行程度排名是由 TIOBE 机构统计给出,每月更新一次。TIOBE 编程开发排行榜是根据互联网上有经验的程序员、课程和第三方厂商的数量,并使用搜索引擎(如 Google、Bing、Yahoo!、百度)以及 Wikipedia、Amazon、YouTube 统计出的排名数据。排名顺序只是反映某个编程语言的热门程度,并不能说明一门编程语言好不好,或者一门语言所编写的代码数量多少。表 1.1 是 2014 年 2 月发布的 TIOBE 编程语言流行程度排行榜。

表 1.1 TIOBE 编程语言流行程度排行榜

2014 年 2 月排名	2013 年 2 月排名	变化	编程语言	占有率	动态变化
1	2	^	C	18.334%	+1.25%
2	1	v	Java	17.316%	-1.07%
3	3		Objective-C	11.341%	+1.54%
4	4		C++	6.892%	-1.87%
5	5		C#	6.450%	-0.23%
6	6		PHP	4.219%	-0.85%
7	8	^	(Visual) Basic	2.759%	-1.89%
8	7	v	Python	2.157%	-2.79%
9	11	^	JavaScript	1.929%	+0.51%
10	12	^	Visual Basic . NET	1.798%	+0.79%
11	16	≈	Transact-SQL	1.667%	+0.89%
12	10	v	Ruby	0.924%	-0.83%
13	9	≈	Perl	0.887%	-1.36%
14	18	≈	MATLAB	0.641%	-0.01%
15	22	≈	PL/SQL	0.604%	-0.00%
16	47	≈	F#	0.591%	+0.42%
17	14	v	Pascal	0.551%	-0.38%
18	36	≈	D	0.529%	+0.23%
19	13	≈	Lisp	0.523%	-0.42%
20	15	≈	Delphi/Object Pascal	0.522%	-0.36%

由表 1.1 可见,C 语言是目前流行程度最广的编程语言,排名前 5 名的编程语言也都与 C 语言相关。另外从 TIOBE 发布的最近 10 年的编程语言的排名变动情况也可看出,C 语言的排名始终位居前列,保持较高的占有率。

C 语言简洁、灵活、高效,独具魅力。如今的程序编写向着越来越冗长、越来越庞大的趋势发展,C 语言则属于相对“低级”的编程语言,其简洁之美无可替代。因此,编程爱好者在众多程序设计语言中格外青睐 C 语言。

C语言高效、通用、使用广泛。支持多种显示器和驱动器,适用多种操作系统以及多种机型,可移植性好,1978年后先后被移植到大、中、小及微型机上;既可用于编写系统程序,也可用于编写不依赖计算机硬件的应用程序。从应用程序到操作系统,从电子表、微波炉到超级计算机、宇宙飞船,从 Windows、Mac OS X、iOS、Linux、Symbian、Android 到 MTK 系统,只要有电子产品的地方,绝大多数都能看到 C 语言的应用。许多系统软件,如 DBASE III PLUS、DBASE IV 都是用 C 语言编写的。

C 语言的发明人,Dennis MacAlistair Ritchie(丹尼斯·里奇,1941. 9. 9—2011. 10. 12)曾说:“C 诡异离奇,缺陷重重,并获得巨大成功。”可见 C 语言并非十全十美,但却获得了巨大成功。

事实上,C 语言由黑客设计、被黑客推崇。当然这里的黑客并非现今人们眼中那些擅自非法进入他人系统从事破坏、攻击、盗取等非法活动之徒,而是真正的英雄、高手,“黑客”一词尚未被人们熟知前,计算机高手们喜欢自称为黑客,并且以被其他黑客承认为荣。他们喜欢挑战技术极限,乐于与他人分享自己的成果,正是这种处处闪耀着的黑客光芒使 C 语言(简称 C)永葆青春。尽管目前程序设计语言众多,但至今为止,C、UNIX、脚本语言仍是年轻黑客被圈子接受前必须苦练的三大技艺,这也是人们在众多的计算机程序设计语言中,为什么要学 C 的又一理由。

1.1.2 C 语言的诞生

C 语言由美国贝尔实验室的 Dennis MacAlistair Ritchie 于 1972 年推出,1978 年由贝尔实验室正式发表。Ritchie 1941 年 9 月 9 日出生于美国纽约,1968 年获哈佛大学数学博士学位。父亲 Alistair E. Ritchie 长期担任贝尔实验室科学家一职,在晶体管理论方面颇有造诣。在父亲影响下,Ritchie 开始走上科学研究之路。

C 的诞生颇为有趣。C 因 UNIX 而诞生,而编写 UNIX 的初衷,是为了潇洒玩游戏。C 的根源可追溯到 1960 年出现的 ALGOL 60 语言,这种高级语言面向问题,离硬件较远,不适合编写系统程序。1963 年剑桥大学将 ALGOL 60 发展成 CPL(Combined Programming Language)语言,CPL 比 ALGOL 60 接近了硬件一些,但 CPL 规模较大,难以实现。1967 年剑桥大学的 Martin Richards 对 CPL 进行简化,产生 BCPL(Basic Combined Programming Language)语言。

1970 年美国贝尔实验室的 Kenneth Lane Thompson(对 Ritchie 的职业生涯影响最大),对 BCPL 进一步简化,设计出 B 语言(简称 B)。B 很简单,只有 8KB,且接近硬件,是一种解释型语言。Thompson 和 Ritchie 用 B 写了 UNIX(一些应用是由 B 语言和汇编语言混合编写)。因 B 过于简单,功能有限,为解决可移植性问题,Thompson 和 Ritchie 对 B 语言进行改进,形成 NB 语言。但在 UNIX 的移植方面,NB 语言依然不尽如人意,此后 Ritchie 对 NB 语言进行改进,于是 C 语言诞生。C 既保持了 BCPL 和 B 语言的优点(精练、接近硬件),又克服了它们的缺点(过于简单、数据无类型等)。1973 年 Thompson 和 Ritchie 两人合作,把 UNIX 的 90%以上用 C 改写。

后来 C 语言经过多次改进,但主要还是在贝尔实验室内部使用。1977 年出现了不依

赖于具体机器的 C 语言编译文本“可移植 C 语言编译程序”,C 的移植工作大为简化,使 UNIX 迅速在各种机器上实现。随着 UNIX 的广泛使用,C 也迅速得到推广。C 和 UNIX 在发展过程中,相辅相成。1978 年后,C 先后被移植到大、中、小、微型机上,如今 C 已风靡全球。C 设计之初,仅作为编写 UNIX 的一种工作语言,后来被众多程序员狂热拥戴,成为使用最广泛的系统开发语言,这是 Ritchie 本人也没预料到的。

1978 年 Brian W. Kernighan 和 Dennis M. Ritchie 合著了 *The C Programming Language*,简称 K&R、K&R 标准、K&R 版 C 教材、“白皮书”等,该书影响深远,成为后来广泛使用的 C 语言版本的基础,称为标准 C。但 K&R 中并没定义一个完整的标准 C 语言。1983 年,美国国家标准化协会 ANSI(American National Standards Institute)根据 C 问世以来各种版本对 C 的发展和扩充,制定了新标准 ANSI C,比标准 C 有很大的发展。目前流行的 C 编译系统大多以 ANSI C 为基础开发,但不同版本 C 编译系统实现的语言功能和语法规则略有差别。

1.1.3 C 语言的特点

C 语言具有以下特点。

- C 集高级语言的成分与低级语言的功能于一身。不仅具有高级语言通常的特点,同时可像汇编语言那样对计算机最基本的工作单元,如位、字节和地址等进行操作。
- C 简洁、紧凑、方便、灵活。仅有 32 个关键字、9 种控制语句。程序书写自由,压缩了一切不必要的成分。
- C 功能齐全。具有多种数据类型、多种运算符、多种类型表达式,灵活运用各种运算符,可实现其他高级语言中难以实现的运算。
- C 的执行效率高。因 C 可直接和计算机硬件打交道,在需要对硬件进行操作时,用 C 写代码,明显优于用其他解释型高级语言,C 是目前执行效率最高的高级语言。
- 语法限制不严,程序设计自由度大。
- C 适用范围广,适合多种操作系统与多种机型。

1.1.4 C 语言源程序的结构

按 C 语言语法规则编写的程序称 C 语言源程序,简称 C 源程序,具有如下结构特征。

- ① 一个 C 语言源程序由一个或多个源文件组成。一个源文件由一个或多个函数组成。
- ② 每个源程序,必须有一个且只能有一个 main 函数,也称主函数。
- ③ 源程序中可以有预编译命令,预编译命令通常放在源文件或源程序的最前面。
- ④ 每一个语句都必须以分号(;)结尾。但预处理命令后、函数头之后不能加分号。
- ⑤ 标识符、关键字之间需加空格(一个或多个)隔开,若已有明显的间隔符,可不加

空格。

⑥ `/* ..... */`或`//.....`为注释语句,仅起注释作用,程序运行时不被执行。前一种形式的注释语句可位于程序任何部位,但后一种形式的注释语句只能放在行末。若注释语句需分开写在两行(或两行以上或多行),则只能用`/* ..... */`。注意,`/*、*/`不能写为`/ *、* /`。另外,有些编译器(例如 TC)不识别后一种形式的注释语句。

⑦ C 源程序中,英文字母区分大小写(例如,主函数名为 `main`,而非 `Main`)。

⑧ 若需要,几个语句可写在同一行,一个语句也可分开写在多个行。

用 C 编程时,必须严格遵循以上规则,无论程序多么复杂或多么简单。此外,和所有程序设计语言一样,C 也有一个基本字符集。C 的标识符、关键字、语句和标准库函数等都是由 C 规定的基本字符组成,只有基本字符集中的字符才可出现在 C 语言源程序中。

C 语言的基本字符集,采用的是 ASCII 字符集。C 的基本字符包括:英文大、小写字母(A~Z、a~z),阿拉伯数字(0~9),运算符(+、-、*、/、%、=、<、>、<=、>=、!=、==、<<、>>、&、|、&&、||、^、()、[]、->、..、!、?:、:,、;、_ ,详见附录),特殊字符(一些具有特定含义的符号,如 {、}、/ *、* /、//等),分隔符(空白符和,)。

空白符是空格符、制表符、换行符等的统称,在字符常量和字符串常量中起空白字符作用,而在其他地方出现时,只起间隔作用,编译程序对它们忽略不计。故程序中,用或不用空白符皆可之处,是否使用空白符,并不影响程序的编译。但程序中正确使用空白符不仅是程序正确之所需,同时可增加程序的清晰性和可读性,而错误使用空白符将导致程序出错(就像学习英语书写句子时,空格符的使用一样)。分隔符,(半角英文)则主要用于分隔同类项,例如用“`int x,y,max;`”定义三个变量,皆为整型。

以下代码段 `c1-1-1.c` 就是一个 C 语言源程序,其中只有一个 `main` 函数。

【例 1.1】 仅有一个主函数 `main` 的 C 源程序。

```
/* 代码段 c1-1-1.c */
#include <stdio.h>           /* 预编译命令 (详见第 9 章) */
main()                      //函数 main 的头部 (详见第 6 章)
{
    printf("Only one!\n");    //函数体: 输出结果 (printf 详见第 3 章)
}
```

以下代码段 `c1-2-1.c` 也是一个 C 源程序,它由两个函数: `main` 和 `volume` 组成。

【例 1.2】 由两个函数组成的 C 源程序。

```
//代码段 c1-2-1.c
#include <stdio.h>           //预编译命令
#define PI 3.14159          //预编译命令: 定义符号常量 PI (详见第 9 章)
main()                      //函数头
{
    int r;                  //函数体的第 1 条语句: 定义变量 (详见第 2 章)
    float L,S,V;           //函数体的第 2 条语句: 定义变量
    float volume(int r);    //函数体的第 3 条语句: 声明函数 volume (详见第 6 章)
```



```

scanf("%d",&r);          //函数体的第4条语句:给变量r赋值(详见第3章)
if(r<0)                  //函数体的第5条语句:if-else语句(详见第4章)
    printf("\n半径r为负,无意义!\n");    //输出(详见第3章)
else
{
    L=2*PI*r;            //对变量L赋值(详见第2章)
    S=PI*r*r;            //对变量S赋值
    V=volume(r);         //调用函数volume(详见第6章)
    printf("\n周长 L=%f,面积 S=%f,体积 V=%f,\n",L,S,V); //输出结果(详见第3章)
}
}

float volume(int r)      //函数头
{
    float c;
    c=PI*4/3*r*r*r;
    return c;
}

```

关于C的源程序(以c1-1-1.c、c1-2-1.c为例),再做以下补充说明。

① C规定,函数由函数头和函数体构成。函数体由若干条语句组成,函数体内若需要定义相关变量或声明相关函数,必须集中在函数体的最前面进行,之后才是若干执行语句。例如c1-1-1.c函数体中,没有变量定义与函数声明部分,仅有一个执行语句。而c1-2-1.c的函数体中,前三条语句为变量定义及函数声明,第三条语句之后的语句皆为执行语句,前三条语句的先后顺序无规定,只要集中位于函数体前部即可,但它们中任何一条皆不能位于执行语句之后,否则出错(复合语句内定义变量例外,详见第4章)。例如,若将c1-2-1.c函数体的第3、4条语句位置对调,则出错(对c1-2-1.c编译,系统将提示出错)。

② 在c1-2-1.c中,不可用 π 来表示圆周率3.14159,因 π 是希腊字母,而C的基本字符集中没有希腊字母。例中用基本字符集中的英文字母P、I组成标识符PI,通过预编译命令,将PI定义为符号常量,代表圆周率。

1.1.5 C源程序的运行

首先,创建C源文件,即对C源程序进行编辑产生.c文件。C源程序可由记事本等编辑程序输入,或在C的集成开发环境,如VC++ 6.0中编辑(更多是在后者中编辑,关于VC++ 6.0请见附录C),C源程序通常以c为扩展名。

然后,运行C源程序。C是编译语言,要运行C源程序,首先需将.c文件,翻译成扩展名为.obj的目标程序(机器指令程序),把高级语言源程序翻译成目标程序的过程称编译。目标程序仍不能被计算机执行。因为用户在编写源程序时,会调用系统库函数,其代码是系统提供而非用户所写的,编译只是将源程序代码转变为机器可识别的二进制代码,要运行程序,必须对调用的库函数代码进行连接,同时对所有机器指令程序进行重定位,