

谭浩强 主编

C语言程序设计 实用教程

Applied
Technique

周虹 郑佳昕 闫瑞华 等编著



免费提供
电子教案



机械工业出版社
CHINA MACHINE PRESS



高等院校计算机应用技术系列教材

C 语言程序设计实用教程

周虹 郑佳昕 闫瑞华 等编著
贾宗福 主审



机械工业出版社

本书共 13 章, 内容包括程序设计基础、简单的数据类型、运算符和表达式、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、编译预处理、指针、结构体与共用体、位运算、文件、实用项目开发技术。书中的程序均在计算机上调试通过。

本书文字严谨、流畅, 例题丰富, 文档规范, 注重程序设计技能训练。

本书可作为高等院校非计算机专业学生学习 C 语言程序设计的教材, 也可作为程序设计爱好者学习 C 语言程序设计的参考书。

图书在版编目(CIP)数据

C 语言程序设计实用教程/周虹等编著. —北京: 机械工业出版社, 2010. 8
(高等院校计算机应用技术系列教材)

ISBN 978-7-111-31398-4

I. ①C… II. ①周… 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(2010)第 143988 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 赵 轩

责任印制: 杨 曦

北京双青印刷厂印刷

2010 年 8 月第 1 版·第 1 次印刷

184mm×260mm·20.5 印张·504 千字

0001—3000 册

标准书号: ISBN 978-7-111-31398-4

定价: 33.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

门户网: <http://www.cmpbook.com>

销售一部: (010) 68326294

教材网: <http://www.cmpedu.com>

销售二部: (010) 88379649

读者服务部: (010) 68993821

封面无防伪标均为盗版

序 言

进入信息时代,计算机已成为全社会不可或缺的现代工具,每一个有文化的人都必须学习计算机,使用计算机。计算机课程是所有大学生必修的课程。

在我国 3000 多万大学生中,非计算机专业的学生占 95% 以上。对这部分学生进行计算机教育将对影响今后我国在各个领域中的计算机应用的水平,影响我国的信息化进程,意义是极为深远的。

在高校非计算机专业中开展的计算机教育称为高校计算机基础教育。计算机基础教育和计算机专业教育的性质和特点是不同的,无论在教学理念、教学目的、教学要求、还是教学内容和教学方法等方面都不相同。在非计算机专业进行的计算机教育,目的不是把学生培养成计算机专家,而是希望把学生培养成在各个领域中应用计算机的人才,使他们能把信息技术和各专业领域相结合,推动各个领域的信息化。

显然,计算机基础教育应该强调面向应用。面向应用不仅是一个目标,而应该体现在各个教学环节中,例如:

教学目标:培养大批计算机应用人才,而不是计算机专业人才;

学习内容:学习计算机应用技术,而不是计算机一般理论知识;

学习要求:强调应用能力,而不是抽象的理论知识;

教材建设:要编写出一批面向应用需要的新教材,而不是脱离实际需要的教材;

课程体系:要构建符合应用需要的课程体系,而不是按学科体系构建课程体系;

内容取舍:根据应用需要合理精选内容,而不能漫无目的地贪多求全;

教学方法:面向实际,突出实践环节,而不是纯理论教学;

课程名称:应体现应用特点,而不是沿袭传统理论课程的名称;

评价体系:应建立符合培养应用能力要求的评价体系,而不能用评价理论教学的标准来评价面向应用的课程。

要做到以上几个方面,要付出很大的努力。要立足改革,埋头苦干。首先要在教学理念上敢于突破理论至上的传统观念,敢于创新。同时还要下大功夫在实践中摸索和总结经验,不断创新和完善。近年来,全国许多高校、许多出版社和广大教师在这领域上作了巨大的努力,创造出许多新的经验,出版了许多优秀的教材,取得了可喜的成绩,打下了继续前进的基础。

教材建设应当百花齐放,推陈出新。机械工业出版社决定出版一套计算机应用技术系列教材,本套教材的作者们在多年教学实践的基础上,写出了一些新教材,力图为推动面向应用的计算机基础教育作出贡献。这是值得欢迎和支持的。相信经过不懈的努力,在实践中逐步完善和提高,对教学能有较好的推动作用。

计算机基础教育的指导思想是:面向应用需要,采用多种模式,启发自主学习,提倡创新意识,树立团队精神,培养信息素养。希望广大教师和同学共同努力,再接再厉,不断创造新的经验,为开创计算机基础教育新局面,为我国信息化的未来而不懈奋斗!

全国高校计算机基础教育研究会荣誉会长 谭浩强

前 言

随着计算机的普及和社会信息化程度的提高,掌握一门计算机语言已经成为计算机用户必备的技能之一。目前,无论是从事计算机专业工作的人员,还是非计算机专业的人员,都将C语言作为学习程序设计的入门语言。C语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好,既具有高级语言的优点,又具有低级语言的许多特点。本书在编写过程中,力求做到概念准确、内容简洁、由浅入深、循序渐进、繁简适当、题型丰富,注重常用算法的介绍,有助于培养学生设计程序和分析问题的能力。书中全部实例都已上机调试通过。

本书既可作为高等院校本专科学生的教材,也可作为其他计算机应用人员学习高级程序设计语言的参考书。

本书共分13章,内容包括程序设计基础、简单的数据类型、运算符和表达式、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、编译预处理、指针、结构体与共用体及枚举类型、位运算、文件、实用项目开发技术简介。

本书第1章和第8章由郑佳昕编写,第2章由薛佳楣编写,第3章由富春岩编写,第4章由杨绪华编写,第5章由王琼编写,第6章由王蕊编写,第7章由闫瑞华编写,第9章由张运香编写,第10章由李兴霞编写,第11章由柏嘉凝编写,第12章由周虹编写,第13章由吕成编写,最后由周虹教授统稿,哈尔滨学院的贾宗福教授担任主审。

程序设计是一门实践性很强的课程,不可能只凭借听课和看书就能掌握,应当重视自己动手编写程序和上机运行程序,上机实践多多益善。为了帮助同学们学习本书,编者还编写了一本配套的学习指导书《C语言程序设计实践教程》,提供本书中各章的学习指导、实验、习题及参考答案。

在本书的编写过程中得到了机械工业出版社和佳木斯大学很多老师的帮助,哈尔滨学院的贾宗福教授审阅了此书,并提出了许多宝贵意见,在此表示衷心的感谢。

编者水平有限,书中难免有疏漏和不足之处,恳请读者提出宝贵意见和建议。

编 者

目 录

序

前言

第1章 程序设计及C语言概述 1

1.1 C语言的发展及特点 1

1.1.1 C语言出现的历史背景 1

1.1.2 C语言的特点 2

1.2 C语言程序的构成 3

1.3 程序的书写风格和书写格式 5

1.4 本章小结 5

1.5 练习与提高 5

第2章 C语言的基本数据及其运算 7

2.1 C语言数据类型简介 7

2.2 标识符 7

2.2.1 字符集 7

2.2.2 标识符概述 8

2.2.3 标识符分类 8

2.3 常量与变量 9

2.3.1 常量 9

2.3.2 符号常量 9

2.3.3 变量 9

2.3.4 变量的初始化 9

2.4 整型数据 10

2.4.1 整型数据在内存中的 存储形式 10

2.4.2 整型常量 11

2.4.3 整型变量 11

2.5 实型数据 12

2.5.1 实型常量 12

2.5.2 实型变量 12

2.6 字符型数据 13

2.6.1 字符常量 13

2.6.2 字符串常量 14

2.6.3 字符变量 15

2.7 运算符和表达式 15

2.7.1 C语言运算符简介 16

2.7.2 表达式的求值规则 16

2.7.3 混合运算中的类型转换 16

2.8 算术运算符和算术表达式 18

2.8.1 基本算术运算符 18

2.8.2 算术表达式和运算符的优先级 与结合性 18

2.8.3 自增、自减运算符 18

2.9 赋值运算与赋值表达式 19

2.9.1 赋值运算符 19

2.9.2 类型转换 19

2.9.3 复合的赋值运算符 20

2.9.4 赋值表达式 20

2.10 逗号运算符和逗号表达式 21

2.11 关系运算符和关系表达式 21

2.11.1 关系运算符及其优先次序 21

2.11.2 关系表达式 21

2.12 逻辑运算符及逻辑表达式 22

2.12.1 逻辑运算符及其优先次序 22

2.12.2 逻辑表达式 23

2.13 条件运算符与条件表达式 24

2.13.1 条件运算符与条件表达式 概述 24

2.13.2 条件运算符的优先级与 结合性 25

2.14 应用举例 26

2.15 本章小结 27

2.16 练习与提高 27

第3章 顺序结构程序设计 30

3.1 C语言语句概述 30

3.2 赋值语句 32

3.3 字符数据的输入与输出 33

3.3.1 字符输出函数 33

3.3.2 字符输入函数 35

3.4 格式的输入与输出	35	6.4.1 字符数组的定义和引用	117
3.4.1 格式输出函数	35	6.4.2 字符数组的初始化	118
3.4.2 格式输入函数	43	6.4.3 字符串的输入和输出	118
3.5 应用举例	47	6.4.4 用于字符处理的库函数	120
3.6 本章小结	49	6.5 应用举例	122
3.7 练习与提高	50	6.6 本章小结	127
第4章 选择结构程序设计	53	6.7 练习与提高	128
4.1 if 语句	53	第7章 函数	132
4.1.1 简单 if 语句	53	7.1 模块化程序设计	132
4.1.2 双分支 if 语句	54	7.1.1 模块化程序设计概述	132
4.1.3 多分支 if 语句	55	7.1.2 函数概述	133
4.1.4 if 语句使用说明	58	7.2 函数的定义	135
4.2 if 语句的嵌套	58	7.2.1 无参函数的定义	135
4.3 switch 语句	61	7.2.2 有参函数的定义	135
4.4 应用举例	63	7.2.3 空函数的定义	136
4.5 本章小结	70	7.2.4 函数的返回值	137
4.6 练习与提高	70	7.3 函数的调用	138
第5章 循环结构程序设计	73	7.3.1 函数调用的形式	138
5.1 while 语句	73	7.3.2 函数的声明	139
5.2 do...while 语句	76	7.3.3 函数参数的传递	141
5.3 for 语句	78	7.4 函数的嵌套调用与递归调用	142
5.4 几种循环的比较	83	7.4.1 函数的嵌套调用	142
5.5 循环嵌套	83	7.4.2 函数的递归调用	143
5.6 continue 语句	85	7.5 数组做函数参数	146
5.7 break 语句	86	7.5.1 数组元素做函数实参	146
5.8 应用举例	87	7.5.2 数组名做函数参数	147
5.9 本章小结	95	7.5.3 多维数组名做函数参数	148
5.10 练习与提高	95	7.6 变量的作用域	148
第6章 数组	99	7.6.1 局部变量	148
6.1 数组和数组元素	99	7.6.2 全局变量	150
6.2 一维数组	100	7.7 变量的存储类别	152
6.2.1 一维数组的定义和使用	100	7.7.1 变量的生存期	152
6.2.2 一维数组的初始化	101	7.7.2 局部变量的存储类别	153
6.2.3 一维数组的应用举例	103	7.7.3 全局变量的存储类别	155
6.3 多维数组	111	7.7.4 存储类别小结	157
6.3.1 二维数组的定义和引用	111	7.8 内部函数和外部函数	158
6.3.2 二维数组的初始化	113	7.8.1 内部函数	158
6.3.3 二维数组的程序举例	114	7.8.2 外部函数	158
6.4 字符数组	117	7.9 应用举例	158

7.10	本章小结	161	9.6.2	指向函数的指针变量做 函数参数	202
7.11	练习与提高	162	9.7	返回指针值的函数	203
第8章	编译预处理	164	9.8	指针数组和指向指针的指针	204
8.1	宏定义	164	9.8.1	指针数组的概念	204
8.1.1	不带参数的宏定义	164	9.8.2	指向指针的指针	205
8.1.2	带参数的宏定义	165	9.8.3	main()函数的命令行参数	206
8.2	文件包含	167	9.9	应用举例	207
8.3	条件编译	169	9.10	本章小结	212
8.4	应用举例	170	9.11	练习与提高	213
8.5	本章小结	172	第10章	结构体与共用体	215
8.6	练习与提高	172	10.1	结构体类型及变量的定义	215
第9章	指针	176	10.1.1	结构体概述	215
9.1	相关概念	176	10.1.2	结构体类型的定义	216
9.1.1	变量的地址	176	10.1.3	结构体变量的定义	217
9.1.2	数据的访问方式	177	10.2	结构体变量的初始化和 引用	218
9.1.3	指针和指针变量	177	10.2.1	结构体变量的初始化	218
9.2	指针变量的定义和使用	177	10.2.2	结构体变量的引用	219
9.2.1	指针变量的定义	177	10.3	结构体数组	220
9.2.2	指针变量的初始化和赋值	178	10.3.1	定义结构体数组	220
9.2.3	指针变量的引用	179	10.3.2	结构体数组的初始化	221
9.2.4	指针的运算	180	10.3.3	结构体数组的应用	221
9.3	指针变量做函数参数	181	10.4	指向结构体类型数据的 指针	222
9.4	数组的指针和指向数组的 指针变量	184	10.4.1	指向结构体变量的指针	222
9.4.1	指向数组元素的指针	184	10.4.2	指向结构体数组的指针	223
9.4.2	通过指针引用数组元素	184	10.4.3	结构体变量和指向结构体的 指针做函数参数	225
9.4.3	数组名做函数参数	188	10.5	用指针处理链表	226
9.4.4	指向多维数组的指针与 指针变量	193	10.5.1	链表概述	226
9.5	字符串的指针和指向字符串的 指针变量	197	10.5.2	处理动态链表所需的函数	228
9.5.1	字符串的表示形式	197	10.5.3	链表的基本操作	228
9.5.2	对字符指针变量与字符数组 的使用	198	10.6	共用体	234
9.5.3	字符指针做函数参数	200	10.6.1	共用体变量的引用方式	235
9.6	函数的指针和指向函数的 指针变量	201	10.6.2	共用体类型数据的特点	235
9.6.1	通过函数的指针变量 调用函数	201	10.7	枚举类型	236
			10.8	用typedef定义类型	238
			10.9	应用举例	240

10.10	本章小结	243	13.1.2	文本与图形	275
10.11	练习与提高	244	13.1.3	图形设计	279
第 11 章	位运算	247	13.2	图形模式的初始化	279
11.1	位运算符与位运算	247	13.3	独立图形运行程序的建立	281
11.1.1	按位与运算符	247	13.4	屏幕颜色的设置和清屏	282
11.1.2	按位或运算符	248	13.5	基本画图函数	284
11.1.3	按位异或运算符	248	13.5.1	画点	284
11.1.4	按位取反运算符	249	13.5.2	画线	284
11.1.5	左移运算符	250	13.6	基本图形的填充	287
11.1.6	右移运算符	250	13.6.1	设定填充方式	287
11.1.7	位运算赋值运算符	250	13.6.2	任意封闭图形的填充	289
11.1.8	不同长度的数据进行位	250	13.7	图形操作函数	290
11.2	位段	252	13.7.1	图形窗口操作	290
11.3	应用举例	253	13.7.2	屏幕操作函数	290
11.4	本章小结	255	13.8	图形模式下的文本操作	292
11.5	练习与提高	255	13.8.1	文本的输出	292
第 12 章	文件	258	13.8.2	文本字体、字型和输出方式的	292
12.1	文件的概念	258	13.8.3	用户对文本字符大小的	294
12.2	文件的操作函数	260	13.9	C 语言动画设计	295
12.2.1	文件的打开	260	13.10	菜单设计技术	298
12.2.2	文件的关闭	261	13.10.1	下拉式菜单的设计	298
12.3	文件的检测函数	261	13.10.2	选择式菜单的设计	300
12.4	常用的读/写函数	262	13.11	大型程序开发的项目管理	301
12.4.1	读/写字符函数	262	13.11.1	项目管理器	301
12.4.2	读/写字符串函数	264	13.11.2	用项目管理器开发程序	301
12.4.3	读/写数据块函数	265	13.11.3	项目管理器的使用技巧	302
12.4.4	格式化读/写函数	268	13.12	本章小结	303
12.5	文件的定位	268	13.13	练习与提高	304
12.5.1	rewind() 函数	269	附录 A	ASCII	306
12.5.2	随机读/写和 fseek() 函数	269	附录 B	运算符和结合性	307
12.6	应用举例	271	附录 C	C 语言常用语法提要	308
12.7	本章小结	273	附录 D	C 语言库函数	312
12.8	练习与提高	273	参考文献	317	
第 13 章	实用项目开发技术简介	275			
13.1	C 语言图形功能简介	275			
13.1.1	图形与硬件	275			

第1章 程序设计及 C 语言概述

计算机是 20 世纪最大的发明，它的出现和飞速发展对社会的各个领域都产生了深远的影响。使用计算机语言开发应用程序和解决实际问题科学技术人员应具备的能力。

为了有效地进行程序设计，编写质量高、易读性好的程序，至少应掌握以下 3 个方面的知识：

- 1) 掌握一门高级语言。
- 2) 掌握解题的方法和步骤，即算法设计，它是程序设计的核心。
- 3) 掌握结构化程序的设计方法。

由于算法的概念及流程图的画法和结构化程序设计方法及 C 语言的发展简史及 C 语言的特点，重点介绍 C 语言程序的构成。在计算机文化基础课中已介绍，本书将不再介绍。本章将介绍。

学习目标

- 了解 C 语言的发展简史及 C 语言的特点。
- 掌握 C 语言程序的构成及书写风格，对 C 语言程序有一个初步了解。

1.1 C 语言的发展及特点

1.1.1 C 语言出现的历史背景

C 语言在 20 世纪 70 年代初诞生于美国的贝尔实验室。在此之前，人们编写系统软件主要使用汇编语言。汇编语言编写的程序依赖于计算机硬件，其可读性和可移植性都比较差，而高级语言的可读性和可移植性虽然与汇编语言相比较好。但一般高级语言又不具备低级语言能够直观地对硬件实现控制和操作，程序执行速度相对较快的优点。在这种情况下，人们迫切需要一种既具有一般高级语言特性，又具有低级语言特性的语言。于是 C 语言就应运而生了。

由于 C 语言兼具高级语言和低级语言的特点，因此迅速普及，成为当今最有发展前途的计算机高级语言之一。C 语言既可以用来编写系统软件，也可以用来编写应用软件。现在，C 语言广泛地应用在机械、建筑和电子等行业，用来编写各类应用软件。

C 语言的发展历程如下。

- 1) ALGOL60：一种面向问题的高级语言。ALGOL60 离硬件较远，不适合编写系统程序。
- 2) CPL (Combined Programming Language, 组合编程语言)：一种在 ALGOL60 基础上更接近硬件的一种语言。CPL 规模大，实现困难。
- 3) BCPL (Basic Combined Programming Language, 基本的组合编程语言)：对 CPL 进行简化后的一种语言。

4) B 语言: 对 BCPL 进一步简化所得到的一种很简单且接近硬件的语言。B 语言取 BCPL 语言的第一个字母。该语言诞生后, UNIX 开始用 B 语言改写。

5) C 语言: 在 B 语言基础上增加数据类型而设计出的一种语言。C 语言取 BCPL 的第二个字母。C 语言诞生后, UNIX 很快用 C 语言改写, 并被移植到其他计算机系统。

6) 标准 C、ANSI C、ISO C: C 语言的标准化。

最初 UNIX 操作系统是采用汇编语言编写的, B 语言版本的 UNIX 是第一个用高级语言编写的 UNIX。在 C 语言诞生后, UNIX 很快用 C 语言改写。C 语言良好的可移植性很快使 UNIX 从 PDP 计算机移植到其他计算机平台。随着 UNIX 的广泛应用, C 语言也得到推广。从此 C 语言和 UNIX 像一对孪生兄弟, 在发展中相辅相成, 很快风靡全球。

从 C 语言的发展历史可以看出, C 语言是一种既具有一般高级语言特性 (ALGOL60 带来的高级语言特性), 又具有低级语言特性 (BCPL 带来的接近硬件的低级语言特性) 的程序设计语言。C 语言从一开始就用于编写大型、复杂的系统软件, 当然也可以用来编写一般的应用程序。

IBM 微机 DOS、Windows 平台上常见的 C 语言版本有:

1) Borland 公司的 Turbo C、Turbo C++、Borland C++ 及 C++ Builder (Windows 版本)。

2) Microsoft 公司的 Microsoft C 及 Visual C++ (Windows 版本)。

1.1.2 C 语言的特点

C 语言以其简洁、灵活、表达能力强、产生的目标代码质量高、可读性强和可移植性好为基本特点而著称于世。特点如下:

1) 程序紧凑、简洁、规整。使用一些简单规则和方法就可以构成相当复杂的数据类型、语句和程序结构。

2) 表达式简练、灵活、实用。C 语言有多种运算符、多种描述问题的途径和多种表达式求值的方法, 这使程序设计者有较大的主动性, 并能提高程序的可读性、编译效率及目标代码的质量。

3) 具有与汇编语言很相近的功能和描述问题的方法。如地址计算、二进制数位运算、使用寄存器存放变量及对硬件端口直接操作等, 都充分利用了计算机系统资源 (如 BIOS 软中断和 DOS 的系统功能调用等)。

4) 具有丰富的数据类型。在系统软件中, 特别是操作系统中, 对计算机的所有软件、硬件资源要实施管理和调度, 这就要求有相应的数据结构作为操作基础。C 语言具有 5 种基本的数据类型: char (字符型)、int (整型)、float (浮点单精度型)、double (浮点双精度型)、void (无值型) 和多种构造数据类型 (数组、指针、结构体、共用体和枚举)。如指针类型使用十分灵活, 用它可以构成链表、树和栈等。指针可以指向各种类型的简单变量、数组、结构体、共用体及函数等。

5) 具有丰富的运算符, 多达 40 余种。丰富的数据类型与众多运算符相结合, 使 C 语言具有表达灵活和效率高的优点。

6) C 语言是一种结构化程序设计语言, 特别适合大型程序的模块化设计。C 语言具有编写结构化程序所必需的基本流程控制语句。保留程序是由函数集合构成的, 函数各自独立作为模块化设计的基本单位。程序所包含的源文件可以分割成多个源程序, 分别对其进行编

译,然后连接起来,构成可执行的目标文件。它提供了丰富的库函数,包括图形函数等,可供用户调用,允许用户根据需要自定义函数。C语言还提供了多种存储属性,可以使数据按其需要在相应的作用域内起作用。

7) 为字符、字符串、集合和表的处理提供了良好的基础,能够表示和识别各种可显示及起控制作用的字符,也能区分和处理单个字符与字符串。

8) 具有预处理程序和预处理语句,给大型程序的编写和调试提供了方便。

9) 程序具有较高的可移植性。在C语言的语句中,没有依赖于硬件的输入/输出语句。程序的输入/输出功能是通过调用输入/输出函数实现的,而这些函数是由系统提供的独立于C语言的程序模块,从而便于在硬件结构不同的计算机之间实现程序的移植。

10) C语言是处于汇编语言和高级语言之间一种中间型的记述性程序设计语言。C语言既具有面向硬件和系统,像汇编语言那样可以直接访问硬件的功能,又有高级语言面向用户,容易记忆、便于阅读和书写的优点。

综上所述,C语言是一种功能很强的语言。但是,它也有一些不足之处:语法限制不严谨,虽然熟练的程序员编程灵活,但安全性低;运算符丰富,完成功能强,但难记、难掌握。因此,学习和使用C语言不妨先学基本部分,熟练后再学习语法规则,进而全面掌握C语言。

1.2 C语言程序的构成

下面先介绍几个简单的C语言程序,然后从中分析C语言程序的构成。

【例1-1】

```
#include <stdio. h>          /*“文件包含”命令*/
main()                      /*求两个数的乘积*/
{
    int a,b,c;              /*定义变量*/
    a=2;b=3;c=a+b;          /*3个赋值语句*/
    printf("%d\n",c);        /*输出c的值*/
}
```

程序的执行结果为:5。

本程序的功能是求两个数a与b的和c。/*...*/是注释,“int a, b, c;”是定义变量。C语言要求程序中用到的变量都要定义。“a=2; b=3; c=a+b;”是3条赋值语句,使a和b的值分别为2和3;c的值为a+b的值。“printf("%d\n",c)”是输出c的值。%d是输入/输出的格式字符串,表示以十进制整型数的输出格式。

【例1-2】

```
#include <stdio. h>          /*“文件包含”命令*/
main()                      /*主函数*/
{
    int a,b,c;              /*定义变量*/
    printf("input two number"); /*提示输入两个数*/
    scanf("%d,%d",&a,&b);      /*输入变量a和b的值*/
    c=mul(a,b);             /*调用函数mul,将得到的值赋给c*/
    printf("product=%d\n",c); /*输出积c的值*/
}

int mul(int a,int b)        /*定义mul函数,函数结果类型为整型,两个形参为整型*/
```

```

int c;          /* mul 函数中的声明部分,定义本函数中用到的变量 c */
c = a * b;      /* a 乘以 b 的值赋给 c */
return c;       /* 将 c 的值返回,通过 mul 带回调用处 */

```

这个程序同样是求两个数的和,但用了两个函数来实现,即主函数和被调用函数 add。mul 函数的作用是将 a 和 b 的和值赋给变量 c, return 语句将 c 的值返回主函数 main。返回的值是通过函数名 mul 带回到 main 函数的调用处。

通过以上两个例子可以知道如下几点。

1) C 语言程序是由函数构成的。一个 C 语言源程序至少包含一个 main() 函数,也可以包含一个 main() 函数和若干个其他函数。在 C 语言中,函数是程序的基本单位。被调用的函数可以是系统提供的库函数(如 scanf() 函数和 printf() 函数),也可以是用户自定义的函数(如例 1-7 中的 add() 函数)。C 语言中的函数相当于其他语言中的子程序,编写 C 语言程序就是编写一个个函数。C 语言的库函数十分丰富,ANSI C 提供了 100 多个库函数。如果要调用库函数,则要用#include 命令把包含库函数的头文件包含在此程序中。

2) 一个函数由两部分组成:一部分是函数首部,另一部分是函数体。
函数首部,即函数的第一行。包括函数类型、函数名、函数的形参、形参类型及函数属性等。例如,例 1-2 中 mul() 函数的首部为

```

int      mul(      int      a,      int      b)
函数类型  函数名   形参类型  形参    形参类型  形参

```

函数名后是函数的参数表,必须用一对小括号括起来。参数表中也可以没有参数,没有参数时也必须要有括号,如 main()。

函数体,即函数首部下面的大括号内的部分。如果一个函数内有多个大括号({}),则最外层的一对大括号为函数体。

函数体一般包括如下两部分。

- 声明部分:在这部分定义变量、对调用函数进行声明等。
- 执行部分:由若干语句组成。

函数的一般格式是

```

数据类型  函数名(函数参数表)
{
    声明部分
    执行部分
}

```

当然,在某些情况下可以没有声明部分,也可以没有执行部分。如:

```

max(){}

```

这是一个空函数,却是合法的。

3) 不管 main() 函数在程序的什么位置,一个 C 语言程序总是从 main() 函数开始执行。

4) C 语言程序的书写格式自由,一行内可以写多个语句,一个语句也可以写在多行上。

C 语言程序没有行号。

5) 每一个语句和数据定义的最后都必须有一个分号,分号是语句必要的组成部分,允许有空语句。空语句只有分号,没有其他内容。

6) C 语言本身没有输入/输出语句,输入/输出由库函数来完成。

7) 可以用/*...*/对 C 语言程序注释,“/”和“*”之间不允许留空格。注释部分可

以出现在程序的任何位置上，可以为若干行。

8) 一个 C 语言程序可以由一个文件组成，也可以由若干个文件组成。一个文件可以包含一个函数，也可以包含多个函数。例 1-2 中包含两个函数，也可以将两个函数放在两个文件中（每个文件有一个文件名）分别进行编译，然后通过连接（link）把它们合成一个可执行文件，以供运行。

总之，C 语言程序可以由若干个源文件构成，每个源文件可以包含编译预处理命令和一个或多个函数，每个函数由函数首部和函数体构成。

1.3 程序的书写风格和书写格式

程序的书写风格直接影响到程序的可读性，对于程序设计具有关键作用。好的设计风格不但可以提高程序设计的质量，而且可以提高程序设计的效率。

- 1) 程序所采用的算法要尽量简单，符合一般人的思维习惯。
- 2) 标识符的使用尽量采取“见名知义，常用从简”的原则。
- 3) 为了清晰地表现出程序的结构，最好采用锯齿形的程序格式。
- 4) 可以用 `/*...*/` 注释，以增加程序的可读性。
- 5) 最好在输入语句之前加一个输出语句，对输入数据加以提示。
- 6) 函数首部的后面和编译预处理的后面不能加分号。
- 7) C 语言程序的书写格式虽然自由，但为了清晰，一般在一行内写一个语句。

1.4 本章小结

本章介绍了算法的概念、流程图的画法、结构化程序设计方法及方法、C 语言的发展简史、C 语言的特点、C 语言程序的构成及编程环境——Turbo C 2.0 和 Visual C++ 6.0 的使用。

重点掌握 C 语言程序的上机操作方法，它是编辑、编译程序及调试程序的实验环境，包括：新建文件、保存文件、打开文件、编译、运行和查看运行结果等。

本章给出了几个简单的 C 语言程序，对于初学者可以在实验课上将其输入，查看运行结果是否与例题中所给出的结果相同。在输入程序的时候，要特别注意程序的基本结构。一般初学者在输入程序时易犯错误，如缺少分号、遗漏大括号、小括号或者多加大括号等。通过几个例题，重点掌握 C 语言程序的构成及程序的书写风格，从开始就养成良好的编程习惯。

1.5 练习与提高

一、单选题

1. 以下不正确的概念是_____。
 - A. 一个 C 语言程序由一个或多个函数组成
 - B. 一个 C 语言程序必须包含一个 main 函数
 - C. 在 C 语言程序中，可以只包括一条语句

D. C 语言程序的每一行上可以写多条语句

2. 下面源程序的书写格式不正确的是_____。

A. 一条语句可以写在几行上

B. 一行上可以写几条语句

C. 分号是语句的一部分

D. 函数的首部必须加分号

3. 在 C 语言程序中_____。

A. main 函数必须放在程序的开始位置

B. main 函数可以放在程序的任何位置

C. main 函数必须放在程序的最后

D. main 函数只能出现在库函数之后

4. C 语言程序的开始执行点是_____。

A. 程序中第一条可以执行的语句

B. 程序中的第一个函数

C. 程序中的 main 函数

D. 包含文件中的第一个函数

二、填空题

1. 一个 C 语言程序由若干个函数构成, 其中必须有一个_____。

2. 一个函数由两个部分组成: 一部分是_____, 另一部分是_____。

3. 注释部分以_____开始, 以_____结束。

4. 在 C 语言中, 构成程序的基本单位是_____。

5. 一个 C 语言程序的开发过程包括: 编辑、_____、连接和运行 4 个步骤。

三、分别用传统流程图和结构化流程图表示以下问题的算法

1. 输入两个数, 交换以后输出。

2. 输入两个数, 按大小输出。

3. 输入 10 个数, 输出它们的乘积结果。

第2章 C语言的基本数据及其运算

数据是程序处理的对象。C语言在程序处理数据之前，要求数据具有明确的数据类型。数据类型是数据结构的表示形式，体现数据的操作属性，对不同数据可进行不同操作。用户在程序设计过程中所使用的每个数据，都要根据其不同用途选择不同类型，一个数据只能有一种类型。在C语言中具有数据属性的量有常量、变量、函数值、函数参数和表达式。

本章将介绍C语言的标识符、基本数据类型、常量、变量、运算符、表达式及其求值。

学习目标

- 了解C语言标识符的构成及正确使用。
- 了解C语言数据类型及基本类型数据的使用。
- 掌握C语言运算符的种类、运算优先级和结合性。
- 掌握C语言表达式类型（赋值表达式，算术表达式，关系表达式，逻辑表达式，条件表达式，逗号表达式）、求值规则及不同类型数据间的转换与运算等。

2.1 C语言数据类型简介

一个程序应包括如下两个方面：

- 1) 对数据的描述。
- 2) 对操作的描述。

不同的数据类型在数据表示形式、合法的取值范围、占用内存空间大小及可参与的运算种类等方面都有所不同。用户在程序设计过程中所使用的每个数据都要根据其不同用途选择不同的类型，每个数据都属于一个确定、具体的数据类型。在C语言中的数据有常量、变量、函数值、函数参数和表达式等，它们分别属于上述类型。

C语言的数据类型十分丰富，如图2-1所示。



图2-1 C语言的数据类型

2.2 标识符

2.2.1 字符集

C语言的字符集是指C语言程序中允许出现的字符，分为以下几类。

- 1) 英文字母（大、小写）：A、B、C、D、…、Y、Z、a、b、c、d、…、y、z。

2) 数字：0、1、2、3、4、5、6、7、8、9。

3) 特殊符号：+、-、*、/、%、=、_、!、(、)、#、\$、^、&、[、]、\、'、"、{、}、|、.、>、<、? 及空格等。

若在程序中使用其他字符，则编译时，会出现语法错误。

2.2.2 标识符概述

C 语言中，处理对象的名称都要用标识符表示，如符号常量名、变量名、函数名、类型名和文件名等。C 语言中的标识符必须满足以下规则：

1) C 语言标识符是由字母、数字、下画线组成的，并且第一个字符必须是字母或下画线。

2) 在 C 语言中，对标识符长度没有统一规定，随系统而不同。

3) 大写字母和小写字母代表不同的标识符，如 SUM 和 sum 是两个不同的标识符。

4) 在使用标识符时，尽量采取“见名知义，常用从简”的原则。如用 max 来代表最大值，这样可以增加标识符的可读性，使程序更加清晰。对于简单数学运算中的变量，用 a、b、c、x、y、z 等简单符号表示。

下面是合法的标识符：

x、xy、_1、x1、x_1、min

下面是不合法的标识符：

11t、\$ t、x.y、a>f、_s#

2.2.3 标识符分类

(1) 关键字

关键字也称为系统保留字，是一类特殊的标识符，在 C 语言中有特殊的含义，不允许作为用户标识符使用，不能用做常量名、变量名、函数名、类型名和文件名等。C 语言中的保留字共 32 个，用小写字母表示。C 语言的保留字如表 2-1 所示。

表 2-1 C 语言的保留字

描述数据类型定义	描述存储类型	描述数据类型	描述语句
typedef	auto	char	break
void	extern	int	case
	register	float	continue
	static	double	default
	volatile	long	if
		short	else
		signed	switch
		struct	for
		union	do
		unsigned	while
		const	return
		enum	sizeof
			goto