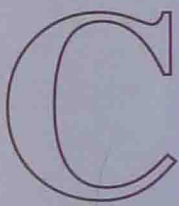


21世纪高等学校计算机基础实用规划教材

C程序设计实用教程



C

周炜 钟红艳 王晓楠 徐云飞 雷晓莉 编著



L



G

清华大学出版社

21世纪高等学校计算机**基础**实用规划教材

C程序设计实用教程

周炜 钟红艳 王晓楠 徐云飞 雷晓莉 编著

清华大学出版社
北 京

内 容 简 介

本书是作者多年以谭浩强先生的《C 程序设计》为教材从事 C 语言教学的经验、应用和研究成果的总结,在很多方面还有所延伸和创新。本书能够加深读者的理解,开阅读者的视野,提高读者的编程能力。本书面向应用,注重算法应用和解题方法,将函数和指针视为家常便饭。全书突出了计算思维的运用。

全书共分 7 章,内容包括 C 语言概述、分支与循环、数组与结构体、指针与链表、递归、文件操作和程序设计竞赛介绍。

本书可以作为国内高校各层次、各专业 C 语言程序设计课程的教材和参考书,也可以作为国家计算机等级考试和国内外各种程序设计大赛的复习、参考和强化训练用书。

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 程序设计实用教程/周炜等编著.--北京:清华大学出版社,2014

21 世纪高等学校计算机基础实用规划教材

ISBN 978-7-302-37321-6

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(2014)第 160013 号

责任编辑:郑寅堃 王冰飞

封面设计:何凤霞

责任校对:时翠兰

责任印制:刘海龙

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:18.5 字 数:454 千字

版 次:2014 年 11 月第 1 版 印 次:2014 年 11 月第 1 次印刷

印 数:1~2000

定 价:35.00 元

产品编号:059208-01

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,为学生的知识、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。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,同一门课程可以有针对不同层次、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 世纪高等学校计算机基础实用规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前言

电子数字计算机和 Internet 的出现缩短了人与人之间的距离,在时间、空间、情感和思维等多个方面改变了人类的生活、生产和思维方式,把人类社会推向了一个崭新的阶段。现在,世界上已经几乎没有人能够避免与计算机直接或间接地打交道了。但计算机并不能理解人们的情感、行为和语言,人们与计算机的“交往”一般是通过计算机的操作系统调度执行各种程序来实现的。而程序是按照一定规则组织起来实现一定功能或完成一定任务的计算机指令集合,是计算机能够“理解”的语言。最常见的程序是 *.exe 文件。

要得到一个计算机程序,首先要用计算机编程语言编写相应的源程序。在计算机编程语言发展的早期,操作系统(如 UNIX、DOS)等系统软件甚至一些应用软件主要是用汇编语言编写的。用汇编程序对一个正确的汇编语言源程序进行汇编、链接就得到一个程序。由于汇编语言是低级编程语言,它直接针对计算机硬件,因此只有非常了解计算机硬件才能编写汇编语言程序。这样一来,用汇编语言所编写的系统软件和应用程序的可读性和可移植性都比较差,而且程序编写和维护的难度也很大,因此汇编语言不能为广大计算机应用人员和计算机编程人员所普遍接受。于是各种高级编程语言(高级语言)就应运而生了。人们熟知的高级语言有 BASIC 语言、FORTRAN 语言、PASCAL 语言、C 语言和 C++ 语言等,现在更有 Visual BASIC、Visual C++、C# 和 Java 等可视化的面向对象的高级编程语言。

用高级语言编写的源程序也是计算机不能直接执行的。BASIC 语言源程序(程序)的每一次执行必须通过 BASIC 解释程序进行解释,边解释边执行;解释程序的作用就像外交场合的“同声翻译”一样。其他高级语言源程序要经过相应语言的编译系统编译、链接成计算机程序,才能执行;编译系统的作用就像外交场合的“书面翻译”一样,只要翻译一次,得到的程序一般就可以在不同的机器上多次执行。

然而,一般高级语言难以实现汇编语言的某些功能(如地址操作和位操作等)。人们需要一种既具有低级语言优点,又具有一般高级语言优点的高级语言。C 语言正是集二者之优点于一身的当之无愧的高级语言,它是国内外广泛使用的执行效率最高且可以直接操作硬件的计算机编程语言。C 语言既可以用来开发系统软件,也可以用来编写应用软件,甚至在某些场合可以用来代替汇编语言开发硬件驱动程序。因而 C 语言是计算机专业人员和计算机应用人员都应当掌握的一种高级编程语言。

国内最著名的 C 语言教科书莫过于谭浩强先生的《C 程序设计》。以谭先生的著作为蓝本的其他 C 语言教科书也非常多。本书是作者多年以谭先生著作为教材从事 C 语言教学的经验、应用和研究成果的总结,在许多方面还有所延伸和创新。本书能够加深读者的理解,开阅读者的视野,提高读者的编程能力。本书面向应用,将 C 语言的绝大多数理论和语法集中在第 1 章讲解,以便使读者从整体上系统地了解和掌握,而其他章节主要讲述算法应

用和解题方法。由于函数的重要性并为了突出模块化程序设计的思想,本书从第 1 章就开始使用函数,而未将函数单独列章;又由于指针的困难性,本书也从第 1 章就开始使用指针以加强训练;总之,本书将函数和指针视为家常便饭。另外,由于共用体类型和位运算都很少使用,本书没有展开讲解它们,对枚举类型讲述也较少。

全书共分 7 章,其中,第 1 章和第 5 章由周炜执笔,第 2 章由周炜、徐云飞执笔,第 3 章由周炜、雷晓莉执笔,第 4 章由钟红艳执笔,第 6 章和第 7 章由王晓楠执笔,附录 A 由王晓楠制作。全书由周炜策划、修改和统稿。第 1 章集中了 C 语言除文件操作以外的大部分理论和语法,因而值得结合有关章节反复研读和练习。电子课件由徐云飞制作。

本书可以作为国内高校各层次、各专业 C 语言程序设计课程的教材和参考书,也可以作为国家计算机等级考试和国内外各种程序设计大赛的复习、参考和强化训练用书。

在本书中,对“程序”和“源程序”两个术语不加区别,请读者根据上下文理解。

本书大部分内容曾在空军工程大学防空反导学院本科生和大专生中讲授多年。本书以 Turbo C++ 3.0 为主要平台,兼顾 Turbo C 2.0 和 Microsoft Visual C++ 6.0。由于作者水平有限,书中(特别是例程序中)一定还有作者未发现的错误、缺点和纰漏。恳请广大读者批评指正,作者不胜感激!

周 炜

2014 年 6 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言的诞生和发展	1
1.2 C 语言的特点	2
1.3 C 语言的数据类型	4
1.4 基本类型数据的机器码	8
1.4.1 机器码和编码方案	8
1.4.2 整型数据的机器码	9
1.4.3 字符型数据的机器码——ASCII 码	14
1.4.4 实型数据的机器码	15
1.5 标识符、常量、变量和字符串常量	16
1.5.1 标识符	16
1.5.2 常量	17
1.5.3 变量、只读变量和可读写变量	19
1.5.4 字符串常量	25
1.6 地址、指针和动态内存分配	26
1.7 定义新类型名	30
1.8 预处理命令	33
1.8.1 宏定义命令	33
1.8.2 文件包含命令	34
1.8.3 条件编译命令	35
1.9 C 语言的运算、表达式和语句	37
1.9.1 运算及其优先级和结合性	37
1.9.2 各类运算的功能	41
1.9.3 各基本类型数据间的混合运算	49
1.9.4 C 语言的表达式	50
1.9.5 C 语言的语句	51
1.10 C 程序的结构与 C 函数	60
1.10.1 C 程序的结构	60
1.10.2 C 函数	60
1.10.3 值传递和地址传递	63

1.10.4	函数原型	64
1.10.5	主程序带形式参数	65
1.10.6	数组名和函数名做函数参数	66
1.11	C 语言的常用库函数	67
1.11.1	格式输入函数和格式输出函数	67
1.11.2	字符函数	69
1.11.3	字符串处理函数	70
1.11.4	清屏函数 clrscr() 和光标函数 gotoxy(x, y)	71
1.11.5	system(s) 函数	71
1.11.6	数学函数	71
1.11.7	内存管理函数	73
1.11.8	内存比较与填充函数	74
1.11.9	随机函数	74
1.11.10	日期函数和时间函数	74
1.12	变量和函数的存储类别	75
1.12.1	变量的存储类别	75
1.12.2	函数的存储类别	79
1.13	多文件编译	80
1.13.1	用文件包含命令实现多文件编译	80
1.13.2	用工程实现多文件编译	81
1.14	C 语言集成开发环境的使用	83
1.14.1	安装和配置 C 语言系统平台	84
1.14.2	C 语言程序设计的方法和步骤	86
习题 1		95
第 2 章	分支与循环	97
2.1	分支结构程序设计	97
2.1.1	条件运算和条件语句	97
2.1.2	if-else 语句	98
2.1.3	switch 语句	100
2.2	循环结构程序设计	103
2.2.1	while 循环语句	103
2.2.2	do-while 循环语句	106
2.2.3	for 循环语句	108
2.2.4	循环语句综合编程	111
习题 2		116
第 3 章	数组与结构体	118
3.1	数组	118

3.1.1	一维数组	119
3.1.2	杨辉三角形	123
3.1.3	二维数组	124
3.2	用一维数组处理矩阵	129
3.3	顺序表	132
3.3.1	顺序表的查找	132
3.3.2	顺序表的排序	134
3.3.3	顺序表的删除	136
3.3.4	顺序表的插入	138
3.4	结构体	140
3.4.1	结构体类型定义及变量声明	140
3.4.2	结构体数组	141
3.4.3	结构体指针	142
习题 3		144
第 4 章	指针与链表	146
4.1	指针做函数参数	146
4.2	一维数组的指针	150
4.2.1	指针的算术运算	150
4.2.2	数组指针做函数参数	154
4.3	函数的指针	157
4.3.1	函数指针变量	158
4.3.2	函数指针做函数参数	160
4.3.3	函数指针数组	162
4.4	返回指针值的函数	164
4.5	指针数组	166
4.6	指向指针的指针——双指针	170
4.7	指针与多维数组	173
4.7.1	多维数组元素的指针	173
4.7.2	指向一维数组的指针变量	177
4.7.3	指向数组的指针做函数参数	178
4.8	链表	179
4.8.1	链表概述	179
4.8.2	链表的创建、输出和销毁	181
4.8.3	链表的查找	184
4.8.4	链表的插入	187
4.8.5	链表的删除	190
4.8.6	链表的排序	192
4.8.7	链表的综合应用	193

习题 4	199
第 5 章 递归	202
5.1 子程序的直接递归	203
5.2 子程序的间接递归	214
5.3 主程序的递归	215
习题 5	218
第 6 章 文件操作	220
6.1 文件的基本概念	220
6.2 文件操作常用函数	222
6.2.1 文件的打开与关闭	222
6.2.2 文件状态函数	223
6.2.3 文件定位函数	224
6.2.4 字符读写函数	224
6.2.5 字符串读写函数	225
6.2.6 整数读写函数	225
6.2.7 格式化读写函数	225
6.2.8 随机读写函数	225
6.3 文件操作举例	226
习题 6	231
第 7 章 程序设计竞赛介绍	233
7.1 程序设计竞赛综述	233
7.2 程序设计竞赛中的编程技巧	237
7.2.1 基本输入输出格式	237
7.2.2 字符串输入	241
7.2.3 关于输出须要专门说明的几个问题	242
附录 A 常用字符与 ASCII 代码对照表	244
附录 B 习题参考答案	246
参考文献	286

本章旨在使读者对 C 语言有一个全面而又概括的了解和整体的把握,并在某些方面还要有较深入的理解。

1.1 C 语言的诞生和发展

C 语言的诞生和发展经过了一个历史过程。

1960 年出现了一种面向问题的高级语言 ALGOL 60。

1963 年英国剑桥大学在 ALGOL 60 的基础上开发出 CPL (Combined Programming Language) 语言。

1967 年剑桥大学的 Martin Richards 推出了简化的 CPL 语言——BCPL (Basic Combined Programming Language) 语言。

1970 年美国贝尔实验室的 Ken Thompson 设计出了基于 BCPL 的 B 语言。

1972 年至 1973 年间,贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了用于描述和实现 UNIX 操作系统的工作语言——C 语言。

1973 年,K. Thompson 和 D. M. Ritchie 合作用 C 改写了 UNIX(第 5 版)。

1975 年,UNIX 第 6 版发布,使人们普遍注意 C 语言。

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》。

1978 年,B. W. Kernighan 和 D. M. Ritchie 以 UNIX 第 7 版的 C 编译系统为基础,出版了影响深远的名著 *The C Programming Language*,为 C 语言的标准化奠定了基础。

1983 年,美国国家标准协会(ANSI)颁布了 ANSI C。

1987 年,ANSI 又公布了新标准——87 ANSI C。

1990 年,国际标准组织(International Standard Organization,ISO)接受 87 ANSI C 为 ISO 的 C 标准,这就是标准 C。

随着时代的变迁,人们以 ANSI C 为基础开发出了各种版本的 C 语言编译系统。在微型机上使用的 C 语言编译系统主要有 Turbo C、Turbo C++、Microsoft Visual C++、Microsoft C、Quick C、Borland C++ 等,其中最流行的是 Turbo C、Turbo C++ 和 Microsoft Visual C++。

由于 Turbo C++ 的 IDE(Integrated Developing Environment,集成开发环境)编辑界面非常简单友好,操作非常方便,且具有强大的帮助功能,可以随时查看系统提供的详细的语法解释和丰富的例程序,又鉴于国家计算机等级考试和各高等院校的 C 语言教学现在主要以 Microsoft Visual C++(后面简称 VC)或 Microsoft Visual InterDev 为系统平台,本书将

以 Turbo C++ 3.0(后面简称 TC++)为主同时兼顾 VC 和 Turbo C 2.0(以后简称 TC)系统平台。建议读者也使用这三种平台学习 C 语言,同一程序在这三种平台下都编译、链接一下看看区别。

1.2 C 语言的特点

C 语言具有如下 20 个特点:

(1) 允许直接访问物理地址,能直接操作硬件,可用来开发系统软件。C 语言提供了位运算;提供了与汇编语言的接口,允许在程序中直接嵌入汇编语言代码;还提供了与 BIOS 的接口、与 DOS 的接口,容易实现 BIOS 显示驱动服务和磁盘服务、DOS 功能调用和中断处理等功能。因此,C 语言能实现汇编语言的大部分功能,能方便地直接操作硬件,可用来开发系统软件,是最接近汇编语言的高级编程语言。正是由于这个原因,有的教科书把 C 语言称为“中级语言”。

由于 C 语言生成的目标代码质量好、执行效率高,因此可以用来代替汇编语言编写电子商务、工农业生产和科学实验中的过程控制等程序。

(2) 运算多样,表达式丰富。C 语言共有 47 种运算。除了人们熟知的+(加)、-(减)、*(乘)、/(除)等算术运算以及!(逻辑非)、&&(逻辑与)、||(逻辑或)等逻辑运算外,C 语言还提供了模运算(%),一个条件运算(?:),一个数据长度运算(sizeof),一对自增、自减运算(++、--),6 个位运算(~、&、|、^、<<、>>),6 个关系运算(==、!=、<、<=、>、>=)等;甚至连圆括号、方括号、逗号、赋值、强制类型转换等都作为运算处理;另外,还提供了指针的后移、前移和减法等算术运算,从而使 C 的运算极其多样化,表达式类型十分丰富。为方便起见,本书把有值函数调用也看成一种运算,并称为“函数运算”。于是共有 48 种运算,但不是每一种运算都有专门的运算符。

(3) 数据类型丰富灵活,便于数据的描述与存储。C 语言提供整型(含枚举型和字符型)、实型(单精度、双精度和长双精度型)、空类型、指针类型、数组类型、结构体类型、共用体类型等数据类型,能用来实现诸如线性表、栈、队列、树之类复杂的数据结构及其操作。尤其是指针类型数据,使用起来比 PASCAL 更为灵活多样。C 语言允许数值在-128 到 127 之间(或 0 到 255 之间)的整型数据与有符号字符型(或无符号字符型)数据通用,为程序设计提供了很大的灵活性和方便性。

(4) 具有结构化的流程控制语句,又用函数作为程序模块以实现程序的模块化,有良好的结构化、模块化功能。C 语言以函数作为程序的基本模块,又具有 switch 语句、if-else 语句、while 语句、do-while 语句、for 语句等结构化的流程控制语句,使得每一函数内部结构清晰,逻辑关系明确。

(5) 语法限制不太严格,编程自由度大。C 语言对数据类型使用灵活,可在必要时进行强制类型转换;在 C 语言中,数值在-128~127 之间(或 0~255 之间)的整型数据与有符号字符型(或无符号字符型)数据通用;C 语言对数组下标越界不做检查,由编程人员自己检查程序的正确性,不过这也给编程人员带来了一点点困难和风险。

(6) C 语言程序可移植性好,基本上不修改就能用于各种型号的计算机和各种操作系统。

(7) C语言风格简洁、紧凑,使用方便、灵活。C语言只有32个基本关键字,10种控制语句,程序书写形式自由,主要用小写字母表示,压缩了一切不必要的成分。32个基本关键字是:

unsigned	char	enum	auto	register	default	for	sizeof
signed	int	struct	const	break	switch	do	typedef
short	float	union	extern	continue	if	while	volatile
long	double	void	static	case	else	return	goto

其中,signed、short、auto、register、volatile、goto这6个关键字基本上不用或很少使用。C语言的10种控制语句分别是:

while 语句	do-while 语句	for 语句	continue 语句	break 语句
if-else 语句	switch 语句	条件语句	return 语句	goto 语句

其中,goto语句是强制转向语句,在本书中不推荐使用,因而也将不做介绍;while语句、do-while语句、for语句统称为循环语句;break语句只用在switch语句和三种循环语句内部;continue语句只用在三种循环语句内部;case语句和default语句只用在switch语句内部。应当指出,用C语言的条件语句完全可以实现if-else语句的控制功能,因此本书把它也归结为控制语句,在其他教科书上一般认为只有9种控制语句。

(8) 库函数丰富。C语言的各个系统平台提供了大量的库函数,它们的定义一般在某个头文件(*.h)中。只要在程序开头用#include命令包含相应的头文件,就可以在程序中使用其中的库函数。例如,只要在程序中写上#include "stdio.h",就可以在程序中用格式输入函数调用语句“scanf("%d", &a);”从键盘输入一个整数到已声明为int型的变量a中;也可以在程序中用格式输出函数调用语句“printf("%d", a);”把已赋值变量a的值输出到屏幕上。

(9) 可以动态地管理内存。在C语言程序中,可以调用malloc函数来动态地分配内存,也可以随时调用free函数来动态地回收已分配内存。

(10) C语言的变量有可读写变量与只读变量之别,有全局变量与局部变量之别,有时还有外部变量与内部变量之别。

(11) C语言的变量都有数据类型和存储类别两个属性。

(12) 支持多文件编译,便于分工协作。一个C语言程序可以按功能模块组织成一个或两个以上文件,因此容易按功能模块划分将编程任务分配给多个编程人员去完成。

(13) 代码复用率高,便于软件升级。由于C语言允许多文件编译,因此每一个编程人员都可以将自己平时测试通过的程序模块按功能分类写入若干个扩展名为.h或.c的文本文件中,以后在任何程序文件中要用到这些功能模块时,只须把所需功能模块的代码直接复制过来,或者用#include命令包含相应的*.h文件或*.c文件,或者将这些*.h文件或*.c文件组织到当前程序所在的工程中,而不必重写这些功能模块的代码。假如已经在文件mymodules.h中加入了函数max的定义:

```
int max(int x, int y) { if (x > y) return x; return y; }
```

则可以在任何程序文件中加入包含命令“#include "mymodules.h"”而直接使用这个max

函数。假如须要修改以前的某个程序,那么就只须修改某些相应程序文件中相应模块的代码,而一般情况下无须修改其他程序文件。

(14) C 语言采用缓冲文件系统,不支持非缓冲文件系统,对文件的所有读写操作都是用库函数来实现的。这在一定程度上降低了文件操作的难度和复杂性。

(15) C 语言提供了强大的条件编译命令,有助于优化程序代码,减小目标程序规模,提高程序运行效率。

(16) 可用 `typedef` 关键字来定义自己喜欢的类型名,还可用 `#define` 命令来定义符号常量,也可用 `const` 关键字来声明只读变量,从而增强程序的可读性和可维护性。

(17) C 程序的执行从主程序开始,且允许主程序(即 `main()` 函数)和子程序(即用户函数)的递归。

(18) 在 C 语言的标识符中大写的英文字母和小写的英文字母有严格的区别,不得通用;C 语言的关键字、预处理命令、库函数名、系统定义的扩展关键字(如 `asm`、`pascal`、`interrupt` 等)和变量名中出现的英文字母都用小写,系统定义的文件类型名 `FILE` 和枚举常量名(如 `RED`、`GREEN` 等)中出现的英文字母一般都用大写,不得改变。

(19) C 语言可与 PASCAL 语言、PROLOG 语言、BASIC 语言等高级语言进行混合编程。

(20) 在 C 语言中可以嵌入关系数据库结构化查询语言 SQL 来访问关系数据库。

C 语言的以上特点仅仅是本书作者的观点。对于这类问题,参加计算机等级考试和其他各种考试的读者请按有关权威部门提供的考试大纲和复习资料进行答卷。

1.3 C 语言的数据类型

C 语言的所有数据类型可以分为简单类型、构造类型和文件类型三大类。文件在绝大多数教科书上不算数据类型,本书把文件列为数据类型之一。构造类型分为数组类型、结构体类型和共用体类型三类。简单类型可分为基本类型、指针类型和空类型三类。

空类型也就是无类型,关键字为 `void`,主要用于声明无参函数、无值函数和无类型指针变量(`void *`),但决不能声明无类型普通变量(即非指针变量)。

指针类型是 C 语言提供的简单数据类型之一,但涉及指针的编程却并不都是简单的事情。在 1.5~1.6 节将对指针类型数据进行简要的和一般的阐述。

基本类型可分为整型(整数类型)和实型(实数类型)两大类。

整型包括字符型、基本整型、短整型、长整型和枚举类型。短整型基本上不用。

基本整型分为有符号基本整型(`int`)和无符号基本整型(`unsigned int`, `unsigned`);短整型分为有符号短整型(`short int`, `short`)和无符号短整型(`unsigned short int`, `unsigned short`);长整型分为有符号长整型(`long int`, `long`)和无符号长整型(`unsigned long int`, `unsigned long`);字符型也分为有符号字符型(`char`)和无符号字符型(`unsigned char`)。当字符型数据专门用来代表字符或字符串时,可以不区别有符号字符型和无符号字符型。

枚举类型数据是对可列举的少数 `int` 型常量的符号化。在程序中用到枚举类型数据时,一般先定义一个新类型名,然后再用这个新类型名去说明枚举类型变量和函数。

在 TC++ 下,实型数据包括单精度型(`float`)、双精度型(`double`)和长双精度型(`long double`),它们的存储长度和十进制有效数字位数如表 1-1 所示。

表 1-1 TC++ 下实型数据的存储长度和十进制有效数字位数

类 型	float	double	long double
长度(字节数)	4	8	10
二进制比特数	32	64	80
十进制有效数字位数	7	16	18

应当注意,长双精度(long double)型数据的存储长度,在 VC、TC++ 和 TC 下为 10 字节,而不是 16 字节。

由于十进制有效数字位数的限制,在程序中处理实型数据时可能发生舍入误差。

例如,在 TC++ 下,double 型数据仅有 16 位十进制有效数字,从左至右第 17 位以后的数字都是不可靠数字,按 double 型数据存储和输出,表达式 $18 + 3.14159265E+17$ 的值与实型常量 $3.14159265E+17$ 完全相同。可见对绝对值大小相差特别大的实型数据进行加法运算或者对相差特别小的实型数据进行减法运算时,由于有效数字位数的限制,可能有些运算数在程序中根本不起作用。因此,在计算精度要求较高的场合,如果须要对绝对值大小有可能相差太大的两个实型数据进行加减法运算或者对特别接近的两个实型数据进行减法运算,应当预先估计各运算数的范围,然后考虑到底选择 float、double 型还是 long double 型变量,实在没有把握就用 long double 型变量,以减少发生舍入误差的机会。

在 TC 下,float 型和 double 型数据的有效数字位数分别为 6 和 15。

各种数据类型的类型名以及基本类型数据的表示范围和它们在内存中所占的字节数(长度)如表 1-2 所示。

表 1-2 C 语言的数据类型

类 型					类型名	长度	数据范围
简单类型	基本类型	基本整型	有符号基本整型	int	2	$[-32\,768, 32\,767]$	
			无符号基本整型	unsigned	2	$[0, 65\,535]$	
		短整型	有符号短整型	short	2	$[-32\,768, 32\,767]$	
			无符号短整型	unsigned short	2	$[0, 65\,535]$	
		长整型	有符号长整型	long	4	$[-2\,147\,483\,648, 2\,147\,483\,647]$	
			无符号长整型	unsigned long	4	$[0, 4\,294\,967\,295]$	
		枚举类型		enum	2	$[-32\,768, 32\,767]$	
		字符型	有符号字符型	char	1	$[-128, 127]$	
			无符号字符型	unsigned char	1	$[0 \sim 255]$	
		实型	单精度型		float	4	$[1.175\,495 \times 10^{-38}, 3.402\,823 \times 10^{38}]$
			双精度型		double	8	$[2.225\,074 \times 10^{-308}, 1.797\,693 \times 10^{308}]$
	长双精度型		long double	10	$[1.000\,000 \times 10^{-4931}, 1.189\,731 \times 10^{4932}]$		
指针类型				*	2		
空类型				void			

续表

	类 型	类型名	长度	数据范围
构造 类型	数组类型	[]		
	结构体类型	struct		
	共用体类型	union		
文件类型		FILE		

应当注意,各数据类型长度与所使用的系统平台有关,在 TC++ 和 TC 下基本上是一样的,表 1-2 列出的就是在 TC++ 下的数据类型长度。在 VC 下与在 TC++ 下的数据类型长度可能有较大的差异。比如在 VC 下,int 数据类型的长度为 4,而不是 2;指针类型变量的长度为 4,而不是 2;float 数据类型的长度为 8,而不是 4。这是本书以 TC++ 而不是以 VC 为主要平台的原因之一。原因之二是有的 C 程序在 VC 平台下不能编译通过。

这里将把 float 型和基本整型称为比字符型更长的数据类型,把长整型称为比基本整型和字符型更长的数据类型,将 double 型称为比 float 型和整型更长的数据类型,将 long double 型称为比 double 型、float 型和整型更长的数据类型。

实型数据的默认小数位数为 6。实型数据的范围由基本上互为对称的两个区间外加一个单点 0.000 000 组成,表 1-2 中只给出了非零值的绝对值区间,即不超过有效数字位数的非零数据范围。例如,float 型数据,其范围实际上为

$[-3.402\ 823 \times 10^{38}, -1.175\ 495 \times 10^{-38}] \cup \{0\} \cup [1.175\ 495 \times 10^{-38}, 3.402\ 823 \times 10^{38}]$;因此,float 型变量值的变化范围是 $[-3.402\ 823 \times 10^{38}, 3.402\ 823 \times 10^{38}]$,在这个区间以外的实数一律当做正负无穷大($\pm \text{INF}$)值处理,而在区间 $(-1.175\ 495 \times 10^{-38}, 1.175\ 495 \times 10^{-38})$ 内的实数值一律当作 0.000 000 处理,这就是机器零。double 型数据和 long double 型数据的情况与 float 型数据类似。

long double 型数据必须被赋值给 long double 型变量才能使用,并且必须用 %Le 或 %Lf 格式符进行输入或输出。其他基本类型数据必须进行(long double)强制类型转换才能用 %Le 或 %Lf 格式符进行输出。double 型数据可以用 %le、%lf 格式符输出,float 型数据可以用 %e、%f 格式符输出。输出实型数据时可以指定尾数的输出数字位数和小数位数,但不得超过相应类型的有效数字位数,否则输出的数据将可能含有无效数字。

必须强调,某种基本类型数据的范围是指这种基本类型的变量能够正确容纳的数据范围,也就是这种类型变量的值的变化范围。一个常量落在某种类型数据的范围之内,并不意味着这个常量就是这种类型的数据。

例如,常量 0 落在所有基本类型数据的范围之内,但它的类型是 int;又如,常量 -32 768 落在 int 型数据的范围之内,但它的类型是 long。何以见得常量 -32 768 是 long 型? 因为 sizeof(-32 768)的值为 4。

纯粹由简单数据类型构成的复合数据类型是构造数据类型。构造数据类型包括数组类型、结构体类型和共用体类型。纯粹由构造数据类型构成的复合数据类型仍然是构造数据