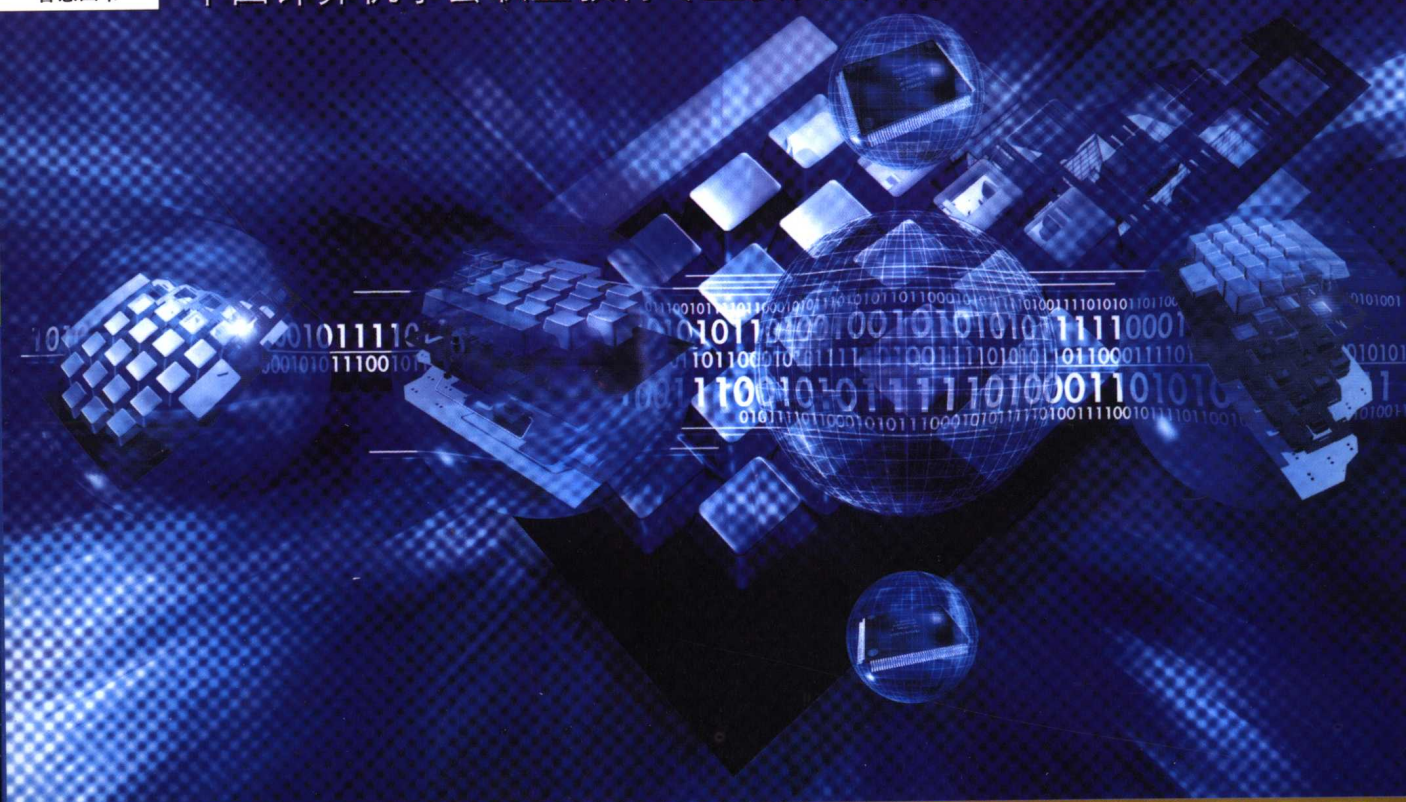




C PROGRAMMING TEXTBOOK

C语言程序设计

基础与应用 (高职高专教材)

编写 / 技能型紧缺人才培养培训教材编写委员会
主编 / 孙振业

内 容 简 介

本书是专为落实教育部和信息产业部《关于确定职业院校开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的通知》和《职业院校计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》精神而编写的标准的职业院校教材。

本书内容：本书由9章和两个附录构成：第1章介绍C语言的简史、特点、基本符号、程序结构、C程序编辑和运行方法、Turbo C系统集成环境的安装、启动和使用；第2章讲解C语言的数据类型、常量、变量、运算符、表达式知识；第3章讲解C语言的算法、常见程序结构和流程等知识；第4章讲解一维、二维、多维数组和字符数组编程技术知识；第5章讲解函数的定义、调用、嵌套、编译、预处理等编程技术；第6章讲解结构类型与联合类型的定义与使用方法、自定义类型的定义、位运算与位字段、移位运算等编程技术与知识；第7章讲解指针、指针变量及利用指针调用数组、函数等各类型数据的编程技术知识；第8章讲解文件的类型基本概念、打开关闭文件的方法、文件的读写与定位等编程技术知识；第9章讲解C语言的实际应用技术；多数章节在讲解编程技术时配有程序举例。

本书特点：1.内容详实、实例丰富、图文并茂、实用性强；2.以边讲解边举例的教学方式，讲解C程序语言的基础知识、实际应用和实际操作，程序实例丰富，方便读者轻松学习，快速上手；3.各章末尾均附有对应本章知识点的练习题，包括填空题、选择题和编程题等几种题型，书后附有答案，更方便教学和自学。

读者对象：高职高专计算机专业教科书，社会C语言培训班教材和广大C语言及计算机编程爱好者的自学读物。

图书在版编目(CIP)数据

C语言程序设计基础与应用/孙振业主编. —北京：海洋出版社，2004.7

全国职业院校技能型紧缺人才培养培训教材. 高职高专

ISBN 7-5027-6145-4

I. C… II. 孙… III. C语言—程序设计—高等学校：技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2004)第059653号

总 策 划：WISBOOK

责任编辑：吴清平 王宏春

责任校对：肖新民

责任印制：肖新民 梁京生

排 版：海洋计算机图书输出中心

出版发行：海洋出版社

地 址：北京市海淀区大慧寺路8号(716房间)

100081

网 址：<http://www.wisbook.com>

经 销：新华书店

发 行 部：(010) 62132549, 62112880-878、875

62174379(传真), 86607694(小灵通)

承 印：北京媛明印刷厂

版 次：2004年7月第1版

2004年7月北京第1次印刷

开 本：787mm×1092mm 1/16

印 张：14.5 彩插1页

字 数：331千字

印 数：1~5000册

定 价：22.00元

本书如有印、装质量问题可与发行部调换

全国高职高专计算机技能型紧缺人才培养培训教材

编 委 会

主 任：吴清平

副主任：程时兴 孙振业 吴振国

委 员（排名不分先后）：

朱诗兵 陶晓欣 卜照斌 李燕萍 邓振杰

周国烛 张建军 王宝智 朴仁淑 宫 谦

韩祖德 周京艳 黄梅琪 王 勇 王宏春

钱晓彬 周珂令

写在前面的话

当前我国正向现代化、信息化、工业化的国家大步迈进,人才资源自然是最为重要的资源。社会各行业、工业企业等部门人才短缺、特别是技能型人才严重短缺,在某种程度上已经影响和妨碍了现代化建设的发展。近年来,我国的职业教育已日益被经济建设所依赖,技能型人才需求存在巨大缺口,因此培养培训任务迫在眉睫。

为配合国家《2003—2007 教育振兴行动计划》,推动“职业教育与培训创新工程”的健康发展,中国计算机学会职业教育专业委员会根据中国计算机学会 2004 年召开的“第八届全国会员代表大会”的精神,努力为落实和推进“职业教育与培训创新工程”做出贡献。

培养培训技能型紧缺人才是职业教育的根本使命和当前的紧迫任务,目的就是要刻不容缓地把这些走进校园的学生培养成适合国家发展和企业需要的有用人才,培养他们成为有一技之长的劳动者和实用型人才,培养的目的主要是就业。

本套教材就是面对目前全国职业院校学生的现状和职业需求而编写的、颇具特色的实用培养培训教材,以配合教育部、劳动与社会保障部、国防科工委、信息产业部、交通部、卫生部联合颁发的《教育部等六部门关于职业院校制造业和现代服务业技能型紧缺人才培养培训工程的通知》而行动。通知中同时颁发了教育部办公厅和信息产业部办公厅《关于确定职业院校开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的通知》,在这个通知中,颁发了《职业院校计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》。

根据以上精神和指导方案,中国计算机学会职业教育专业委员会与海洋出版社海洋智慧图书有限公司,特组织北京、河北、大连、长春、唐山、武汉、深圳、肇庆和杭州等地主要职业院校负责人和一线教师,召开教材研讨会,相互交流经验,介绍需求,共同策划和编写了本套《全国职业院校技能型紧缺人才培养培训教材》,倾心奉献给全国广大的教师和学生,为国家“职业教育与培训创新工程”推波助澜,为满足社会巨大的人才培养需求做出应有的贡献!

整套书的编写宗旨

- 三符合:符合教育部教学大纲、符合市场技术潮流、符合职业院校专业课程需要。
- 技术新、任务明、步骤细致、实用性强,专为技能型紧缺人才量身定制。
- 软件功能与具体范例操作紧密结合,边讲解边动手,学习轻松,上手容易。
- 三适应:适应新的教学理念、适应学生水平现状、适应用人标准要求。

整套书的特色

- 理论精练够用、任务明确具体、技能实操落实,活学活用。

教材编委会

2004 年 7 月

前 言

C语言是优秀的结构化程序设计语言之一，是学习程序设计的必修基础课程。虽然，面向对象的程序设计在今天已经很流行，但其设计理念掌握起来比较困难。为解决这个矛盾，许多高职院校采取了折衷的办法，即先学习C语言，再学习C++面向对象的程序设计。

C语言结构严谨、数据类型完整、语句简练灵活、标准函数众多、运算符丰富，一直受到程序设计人员的好评。

本书推荐授课课时为70~80学时，并提倡实验教学的课时数不低于30学时。本书共由9章组成，其中第1~8章为必修，第9章可以安排学生自修。建议授课教师的教学重点放在第1~8章，即表达式、控制结构与语句、数组、函数，并反复上机训练。如果前5章的内容掌握扎实，后继章节的内容自然容易学习。第6章结构与联合应注重讲授结构数组的应用，而联合与位字段可以选讲。第7章指针应注重讲授指针在函数中的使用，而链表等内容可以选讲。第8章文件中的缓冲型文件是应重点讲授的内容，必须让学生明白，只有文件存储的数据才能永久保存。

本书注重强化编程基础知识与程序设计能力的训练，每一章均安排了精选的习题选解与上机实习的内容，力求通过循序渐进的方法强化学生的编程能力。本书的全部程序均已上机调试通过。

程序设计课程应注重培养学生的逻辑思维能力，提倡严谨的编程风格。建议教师在讲授程序时，适当配以流程图，并辅导学生自己设计流程图。笔者认为，学生学习程序设计应分为三个阶段：第一是读懂简单的程序；第二是根据已有的程序稍加改动使其成为自己的程序；第三是设计流程图并编写程序。在第二阶段时，教师应辅导学生设计流程或给出适当的流程图，以供学生编写程序。上机调试程序、改正程序中的语法错误与逻辑错误是编程训练必不可少的重要环节。

本书由北京信息职业技术学院教师孙振业主编，北京信息职业技术学院教师乐新、黎梅、刘宝芹参加编写。本书编者均是从事计算机编程教学多年的教师。

教材中难免会存在一些缺陷和不足，恳切希望广大读者与教师提出宝贵意见，以便再版时加以改进。

编 者

2004年6月

目 录

第 1 章 C 语言基础	1
1.1 C 语言简史及特点.....	1
1.1.1 C 语言的发展历史.....	1
1.1.2 C 语言的特点.....	1
1.2 C 语言的基本符号与词汇集.....	2
1.2.1 基本符号集.....	2
1.2.2 标识符.....	2
1.2.3 保留字.....	3
1.2.4 ASCII 码字符集.....	3
1.3 C 语言程序的基本结构.....	4
1.3.1 源程序的书写格式.....	4
1.3.2 函数的构成.....	5
1.3.3 C 语言程序的结构特点.....	5
1.4 C 语言程序的编辑及运行.....	6
1.4.1 C 语言程序编辑及运行的一般步骤.....	6
1.4.2 Turbo C 集成环境简介.....	6
1.4.3 安装和启动 Turbo C 系统.....	7
1.4.4 Turbo C 集成环境的使用.....	8
1.4.5 退出 Turbo C 环境.....	9
1.4.6 其他常用菜单选项.....	9
1.4.7 关于 Visual C++ 运行环境.....	10
1.5 习题选解.....	10
1.6 课堂实训 Turbo C 系统的基本操作方法.....	10
1.7 习题.....	11
第 2 章 数据类型及其运算	13
2.1 C 语言的数据类型.....	13
2.2 常量.....	13
2.3 变量.....	15
2.3.1 变量的基本数据类型.....	15
2.3.2 变量的说明.....	15
2.4 运算符和表达式.....	17
2.4.1 运算符的优先级与结合性.....	17
2.4.2 算术运算符与算术表达式.....	18
2.4.3 关系表达式.....	20

2.4.4 逻辑表达式.....	20
2.4.5 其他运算符与表达式.....	21
2.5 习题选解.....	23
2.6 课堂实训 表达式的应用.....	24
2.7 习题.....	25
第 3 章 C 语言的控制结构	28
3.1 算法与结构.....	28
3.1.1 算法的基本概念.....	28
3.1.2 常用流程图.....	29
3.1.3 结构化程序设计的基本概念.....	29
3.2 顺序结构.....	30
3.2.1 赋值语句.....	30
3.2.2 输入输出库函数.....	30
3.2.3 复合语句与空语句.....	33
3.2.4 注释行.....	34
3.3 选择结构.....	35
3.3.1 条件选择结构 if-else.....	35
3.3.2 条件运算符和条件表达式.....	40
3.3.3 开关选择结构 switch-case.....	40
3.3.4 无条件转向语句 goto.....	42
3.4 循环结构.....	42
3.4.1 while 循环.....	43
3.4.2 do-while 循环.....	43
3.4.3 for 循环.....	44
3.4.4 多重循环.....	45
3.4.5 辅助语句.....	46
3.5 应用举例.....	47
3.6 习题选解.....	51
3.7 课堂实训 选择与循环结构的应用.....	54
3.8 习题.....	56
第 4 章 数组	60
4.1 数组的概念.....	60
4.2 一维数组.....	60
4.3 二维数组.....	62
4.4 字符数组.....	65
4.4.1 字符串与字符数组.....	65



4.4.2 字符与字符串函数.....	66	6.5.2 枚举类型的引用.....	116
4.5 应用举例.....	69	6.6 习题选解.....	118
4.6 习题选解.....	72	6.7 课堂实训 结构的应用.....	121
4.7 课堂实训 数组的应用.....	74	6.8 习题.....	122
4.8 习题.....	75	第7章 指针	125
第5章 函数	78	7.1 指针与指针变量.....	125
5.1 函数的定义与调用.....	78	7.1.1 指针的概念.....	125
5.1.1 函数的定义.....	78	7.1.2 指针变量.....	126
5.1.2 函数的调用.....	79	7.1.3 指针变量的运算.....	127
5.1.3 函数的说明.....	82	7.2 指针与数组.....	129
5.2 函数的嵌套调用与递归调用.....	82	7.2.1 指向数组的指针.....	129
5.2.1 函数的嵌套调用.....	82	7.2.2 指针数组.....	131
5.2.2 函数的递归调用.....	83	7.2.3 多级指针.....	132
5.3 变量的存储属性.....	84	7.3 指针与函数.....	133
5.3.1 变量的作用域.....	84	7.3.1 指针作为函数的参数.....	133
5.3.2 变量的存储属性.....	85	7.3.2 返回指针值的指针函数.....	134
5.3.3 内部函数与外部函数.....	87	7.3.3 指向函数的指针.....	135
5.4 编译预处理.....	88	7.3.4 带命令行参数的主函数.....	137
5.4.1 宏定义.....	88	7.4 指针与结构.....	138
5.4.2 文件包含.....	90	7.4.1 结构型指针的定义.....	138
5.5 应用实例.....	90	7.4.2 用结构型指针访问结构成员.....	138
5.6 习题选解.....	96	7.5 链表.....	139
5.7 课堂实训 函数的应用.....	98	7.5.1 链表的构成.....	139
5.8 习题.....	101	7.5.2 链表的常用函数.....	140
第6章 结构与联合	104	7.5.3 链表的基本操作.....	141
6.1 结构.....	104	7.6 应用举例.....	146
6.1.1 结构类型定义.....	104	7.7 习题选解.....	151
6.1.2 结构变量.....	105	7.8 课堂实训 指针应用.....	154
6.1.3 结构数组.....	106	7.9 习题.....	156
6.2 联合.....	108	第8章 文件	160
6.2.1 联合类型的定义.....	108	8.1 文件概述.....	160
6.2.2 联合变量.....	109	8.1.1 文件的概念.....	160
6.3 自定义类型.....	110	8.1.2 文件的分类.....	160
6.4 位运算.....	110	8.1.3 文件的操作.....	161
6.4.1 位运算.....	110	8.2 缓冲型文件输入输出系统.....	161
6.4.2 位字段.....	113	8.2.1 文件类型指针.....	161
6.5 枚举.....	115	8.2.2 文件的打开与关闭.....	162
6.5.1 枚举类型的定义和枚举变 量的说明.....	116	8.2.3 文件的读写.....	163
		8.2.4 文件的定位.....	169



8.3 非缓冲型文件输入 / 输出系统	172
8.3.1 文件的打开与关闭	172
8.3.2 文件的读写	173
8.3.3 文件指针的定位	174
8.4 标准设备文件及 I/O 改向	176
8.4.1 标准设备文件	176
8.4.2 文件的输入输出改向	177
8.5 工程文件	178
8.5.1 工程文件的概念	178
8.5.2 建立多源文件的可执行文件	179
8.5.3 工程文件中的依赖关系	180
8.6 习题选解	180
8.7 课堂实训 文件的应用	181
8.8 习题	182
第 9 章 C 语言应用	184
9.1 文本屏幕输出和文本窗口	184
9.1.1 文本的屏幕输出	184
9.1.2 窗口与显示方式	186
9.1.3 属性控制	187

9.1.4 状态查询函数	188
9.2 菜单技术	190
9.2.1 菜单窗口	190
9.2.2 应用示例	192
9.3 Turbo C 的图形处理	196
9.3.1 图形系统控制	196
9.3.2 颜色控制、画图及填充色	199
9.3.3 图形方式下的文本输出	205
9.3.4 应用示例	206
9.4 C 语言的命令行编译程序 TCC. EXE	209
9.4.1 命令行编译程序 TCC.EXE 的功能	209
9.4.2 命令行编译程序 TCC.EXE 的 调用	209
附录 A 常用字符与 ASCII 代码对照表	211
附录 B Turbo C 常用库函数	212
附录 C Visual C++6.0 编译系统简介	222

第 1 章 C 语言基础

本章要点

- ☑ C 语言简史及特点
- ☑ C 语言程序的构成
- ☑ Main() 函数与其他函数
- ☑ 源程序的书写规则
- ☑ C 语言的运行环境

随着计算机的普及发展, C 语言在各个领域的应用越来越广泛。几乎各类计算机都支持 C 语言的开发环境, 这为 C 语言的普及和应用奠定了基础。

1.1 C 语言简史及特点

1.1.1 C 语言的发展历史

C 语言是一种编译性程序设计语言, 它与 Unix 操作系统紧密地联系在一起。Unix 系统是通用的、交互式的计算机操作系统, 它诞生于 1969 年, 是由美国贝尔实验室的 K.Thompson 和 D.M.Ritchie 用汇编语言开发成功的。

C 语言的前身是 BCPL 语言。1967 年英国剑桥大学的 Martin Richard 推出 BCPL 语言(Basic Combined Programming Language)。1970 年贝尔实验室的 K.Thompson 以 BCPL 语言为基础, 开发了 B 语言, 并用 B 语言编写了 Unix 操作系统, 在 PDP-7 计算机上实现。1972 年贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出 C 语言, C 语言既保持了 BCPL 语言和 B 语言的精练、接近硬件的优点, 又克服了它们过于简单的缺点。1973 年, K.Thompson 和 D.M.Ritchie 合作把 Unix 的 90% 以上用 C 语言改写, 并加进了多道程序设计的功能, 称为 Unix 第 5 版, 开创了 Unix 系统发展的新局面。1975 年 Unix 第 6 版颁布后, C 语言得到计算机界的普遍认可, 从此, C 语言与 Unix 系统一起互相促进, 迅速发展。

最初, 设计 C 语言的目的只是为了描述和实现 Unix 操作系统。目前, C 语言已独立于 Unix 系统, 先后被移植到大、中、小型计算机及微机上。1978 年 B.Kernighan 和 D.M.Ritchie 合作编写了经典著作“*The C Programming Language*”, 它是目前所有 C 语言版本的基础。1983 年美国国家标准化协会(ANSI)对 C 语言问世以来的各种版本进行了扩充, 制定了 ANSI C。现在流行的 C 语言版本有: Microsoft C, Turbo C, Quick C, Borland C 等。本书主要介绍 Turbo C。

1.1.2 C 语言的特点

C 语言具有以下几个基本特点:

(1) C 语言是结构化程序设计语言。C 语言程序的逻辑结构由顺序、选择和循环 3 种基本结构组成, 便于采用自顶向下、逐步细化的结构化程序设计技术。用 C 语言编制的程序, 具



有容易理解、便于维护的优点。

(2) C 语言是模块化程序设计语言。C 语言的函数结构、程序模块间的相互调用及数据传递和数据共享技术,为大型软件设计的模块化分解技术、软件工程技术的应用提供了强有力的支持。

(3) C 语言具有丰富的运算能力。除具有一般高级语言所拥有的四则运算及逻辑运算功能外,还具有二进制的位(bit)运算、单项运算和复合运算等功能。

(4) C 语言具有丰富的数据类型和较强的数据处理能力。不但具有整型、实型、双精度型,还具有结构、联合等构造类型,并为用户提供了自定义数据类型。此外,C 语言还具有预处理能力,能够对字符串或特定参数进行宏定义。

(5) C 语言具有较强的移植性。C 语言程序本身并不依赖于计算机的硬件系统,只要在不同种类的计算机上配置 C 语言编译系统,即可达到程序移植的目的。

(6) C 语言不但具有高级语言的特性,还具有汇编语言的特点。既有高级语言面向用户、容易记忆、便于阅读和书写的优点;又有面向硬件和系统,可以直接访问硬件的功能。

(7) C 语言具有较好的通用性,它既可用于编写操作系统、编译程序等系统软件,也可用于编写各种应用软件。

1.2 C 语言的基本符号与词汇集

1.2.1 基本符号集

任何一种计算机系统所使用的符号是固定的、有限的,它受计算机设备的制约。C 语言就是由一个有限的符号集组成。C 语言的基本符号集采用 ASCII 码字符集,包括:

- (1) 大小写英文字母各 26 个;
- (2) 阿拉伯数字 0~9;
- (3) 特殊符号 39 个,包括运算符和操作符:

+	-	*	/	%	<
<=	>	>=	==	!=	&&
	!	&		~	=
++	--	?:	<<	>>	()
[]	..	->	^	#	sizeof
+=	-=	*=	/=	%=	&=
^=	=	,			

1.2.2 标识符

1. 标识符的作用

在 C 语言中,标识符主要作为常量、变量、函数及自定义类型的名字使用。

2. 标识符的构成规则

(1) 标识符必须以英文字母或下划线开头,并由字母、数字和下划线组成。

例如:abcd, y105, year_day, b20a 都是合法的标识符,而 12agh, a+b-c 则是不合法的。

(2) 大小写字母的含义不同。

例如: asdf, ASDF 和 Asdf 是完全不同的 3 个标识符。

(3) 一个标识符可以由多个字符组成, 但一般只有前 8 个字符有效。

例如: student_a, student_b, student_c 具有相同的意义, 表示同一个标识符 student_。

1.2.3 保留字

保留字又称为关键字, 是 C 语言编译系统所固有的、具有专门意义的标识符。保留字一般用作 C 语言的数据类型名或语句名。C 语言的保留字只有 32 个, 如表 1-1 所示。

表 1-1 C 语言保留字

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

保留字的使用说明:

(1) 所有保留字的字母均采用小写。

(2) 保留字不能再作为用户的常量、变量、函数和类型等的名字。

(3) 在语句中, 应该把保留字与数据或语句的其他部分, 用空格或其他语法允许的专用字符分隔开, 即, 应当对保留字加以适当的分界, 以便于 C 语言编译系统识别它们。

1.2.4 ASCII 码字符集

计算机只能识别二进制代码, 计算机所使用的各种符号是转换为二进制代码进行保存的。在计算机系统中使用的最广泛的是 ASCII 码。

ASCII 码是美国标准信息交换码 (American Standard Code for Information Interchange, 缩写为 ASCII)。ASCII 码虽然是美国国家标准, 但已被国际标准化组织 (ISO) 认定为国际标准。目前, ASCII 码已为世界公认, 并在世界范围内通用。

ASCII 码有 7 位版本和 8 位版本两种, 本书附录一提供了 8 位版本, 共 256 个字符。

7 位版本的 ASCII 码字符集中包含基本字符与控制字符两部分:

基本字符的代码值为 32~127, 为常用字符。

控制字符的代码值为 0~31, 一般是计算机发向外部设备的命令码, 它们仅仅控制外部设



备实现某些特定动作。例如，07 码可以使计算机的蜂鸣器鸣叫。

ASCII 码可用二进制、八进制、十进制和十六进制代码表示。

C 语言中的字符代码采用 ASCII 码表示，C 语言中的 ASCII 码一般采用十进制代码形式。

1.3 C 语言程序的基本结构

1.3.1 源程序的书写格式

简单的 C 语言程序示例：

【程序实例 1.1】 任意输入两个数，输出其中的大者。

```
/*计算两个数的最大值*/
main()                                /*定义主函数*/
{
    float a,b;                        /*说明a、b是实型变量*/
    scanf("%f%f",&a,&b);              /*接收键盘输入的实型数据并赋给变量a和b*/
    if(a>b)                            /*比较a和b的大小*/
        printf("%f\n",a);              /*如果a大，则输出a的值*/
    else
        printf("%f\n",b);              /*否则，如果b大，则输出b的值*/
}
```

程序实例 1.1 是一个简单的 C 语言程序，其中 main() 表示主函数；由大括号 { } 括起来的部分是函数体；/*...*/ 表示对程序或语句的注释；scanf() 和 printf() 是标准输入输出库函数。main() 函数中定义了两个实型变量 a 和 b，并用 scanf() 函数从键盘接收 a 和 b 的值，if-else 是一个条件判断语句，如果 a>b 成立，则用 printf() 函数输出 a 的值，否则输出 b 的值。

【程序实例 1.2】 计算任意输入的 3 个数的平均值。

```
float average(x,y,z)                  /*定义被调函数average的返回值为实型*/
float x,y,z;                          /*定义被调函数average的参数x、y、z为实型*/
{
    float aver;                       /*定义aver为实型变量*/
    aver=(x+y+z)/3;                   /*计算x、y、z的平均值并赋给变量aver*/
    return(aver);                    /*将变量aver的值返回主函数的调用处*/
}
main()                                /*主函数*/
{
    float a,b,c,ave;                  /*定义a、b、c、ave为实型变量*/
    scanf("%f%f%f",&a,&b,&c);          /*从键盘输入变量a、b、c的值*/
    ave=average(a,b,c);               /*调用average()函数并将返回的平均值赋给
变量ave*/
    printf("average=%f",ave);          /*输出变量ave的值*/
}
```

程序实例 1.2 包括两个函数：主函数 main() 和被调用函数 average()。主函数 main() 负责从键盘接收 a、b、c（称实际参数）的值，并传递给 average() 函数的 x、y、z（称形式参数）。Average() 函数计算 x、y、z 的平均值并赋给变量 aver，return 语句将 aver 的值返回给主函数

main()的调用处。

1.3.2 函数的构成

C语言程序由函数组成。一个函数由函数说明部分和函数体两部分构成。

1. 函数的说明部分

函数的说明部分包括函数名、函数类型、参数名、参数类型。

例如：程序实例 1.2 中 average()函数的说明部分如下：

<u>float</u>	<u>average</u>	<u>(x,y,z)</u>
函数类型	函数名	函数参数
<u>float</u>	<u>x,y,z;</u>	
参数类型	参数名	

其中，函数参数的个数与类型可根据需要设立，不需要时可省略。例如 main()函数无参数，而 average()函数有 3 个参数。

函数名、函数参数名的构成应符合标识符的规定。

2. 函数体

函数体是由大括号“{……}”括起来的部分。函数体一般由说明语句与可执行语句两部分组成。

例如：程序实例 1.1 中的“float a,b;”语句是说明语句，其他语句是可执行语句。

当函数体省略时，该函数称为空函数。

例如：a()

```
{ }
```

是一个函数名为 a 的空函数。

关于 C 函数更详细的说明请见第 5 章。

1.3.3 C语言程序的结构特点

从上述程序实例与函数的构成可以看到，C语言程序具有以下结构特点。

1. C语言程序由函数构成

(1) 函数是构成 C 语言程序的基本单位，即 C 语言程序由一个或多个函数组成，但必须有且只能有一个名为 main 的主函数。例如，程序实例 1.1 中只包含一个 main()函数，程序实例 1.2 中则包含两个函数，主函数 main()和被调函数 average()。

(2) C 语言程序总是从 main()函数开始执行，并在 main()函数中结束。这与 main()函数在程序中的位置无关，因此 main()函数可以放在程序的任何位置。

(3) 被调函数可以是系统提供的库函数（例如 scanf()和 printf()函数），也可以是用户根据需要自己编制的函数（例如程序实例 1.2 中的 average()函数）。

(4) 函数名后面必须有一对圆括号“(”和“”)”，这是函数的标志。

(5) 函数体必须由一对大括号“{……}”括起来。一个函数至少有一对大括号，当出现多个大括号时，最外层的一对为函数体的范围。



2. 一个函数由若干行组成

- (1) 一行内可以写多个语句；一个语句也可以写在多行上，并使用“\”作续行符。
- (2) 语句或变量说明的最后必须有一个分号，分号是语句或变量说明的结束标志。
- (3) 为了增强可读性，可以利用/*...*/对 C 语言程序中的任何部分做注释。

3. C 语言程序的书写习惯是使用小写英文字母

下面就请大家注意，书写程序语句一律用小写。

1.4 C语言程序的编辑及运行

本节主要介绍 C 语言程序的处理步骤和 Turbo C 集成环境的使用。

1.4.1 C 语言程序编辑及运行的一般步骤

C 语言采用编译方式将源程序翻译为二进制代码并执行，一般过程分为 4 个步骤。

1. 程序编辑

程序编辑包括以下内容：

- (1) 将源程序输入到计算机内存。
- (2) 修改源程序，并将修改好的源程序存入磁盘。源程序文件的扩展名为.c。

2. 程序编译

C 语言编译程序对源程序进行语法检查，如果发现语法错误时，将在屏幕上显示出错信息。此时，应重新编辑、修改源程序，编辑结束后再重新编译。正确的源程序文件经过编译后，形成二进制文件，其扩展名为.obj。

3. 程序连接

编译后产生的二进制文件，不能直接运行，因为每一个程序模块往往是单独编译的。二进制文件需要经过连接，把各个二进制模块以及系统提供的标准库函数等进行连接处理后，形成可执行文件。可执行文件的扩展名为.exe。

4. 程序运行

可执行文件形成后，可以在操作系统支持下运行，从而得到运行结果。

1.4.2 Turbo C 集成环境简介

C 语言有许多种编译器，这些编译器之间只有很小的区别，只要学会其中的一种，对其他几种就能很快适应。本节主要介绍目前在微机上和 DOS 环境下最为流行的 C 语言版本——Turbo C。

Turbo C 是美国 Borland 公司的产品。它是一个集程序建立、编辑、编译、连接、运行于一体的集成开发环境，是一个易学、易用、快速高效、方便用户使用的程序开发工具。Turbo C 具有图形库和文本窗口函数库，具有查错及协处理器的仿真功能。在国内使用比较普遍的是 Turbo C 2.0，它是在 MS-DOS 操作系统上使用的编译系统，也可以在 Windows 98 操作系统下

运行。本书的所有例题都在 Turbo C 环境下调试通过，后面的介绍都以此为基础背景。

1.4.3 安装和启动 Turbo C 系统

1. 安装

Turbo C 2.0 的源程序盘应该包括 6 个部分的内容：

Install/Help（安装程序和帮助文件等）；

Integrated development environment（集成开发环境，包括编译程序和管理工具等）；

Command line/Utilities（命令行编译程序/实用工具，包括预处理程序和链接程序等）；

Libraries（库程序，包括模式启动程序和数学库、运行库、图形库等）；

Header files/Libraries（头文件和标准程序库）；

Examples/Bgimics（实例库文件，包括一些源程序代码等）。

安装时，可按不同需要把源盘文件安装到软盘或硬盘上，安装方法有两种：

(1) 使用拷贝命令，将必需的文件（TC.EXE、*.H、*.LIB、*.OBJ）拷贝到工作软盘上或硬盘的 TC 子目录下。

(2) 运行源盘中的安装程序 INSTALL.EXE 文件，按照提示进行下去，直到安装完毕。

2. 启动

Turbo C 一般安装在 C 盘根目录下的 TC 子目录中，TC 子目录中包含另外两个子目录：LIB 和 INCLUDE。在 LIB 子目录中存放着“库文件”，在 INCLUDE 子目录中存放着 Turbo C 系统的所有“头文件”（即标准库函数）。

首先进入 C 盘的 TC 子目录，然后键入 TC 并按回车键，即：

```
C:\>CD\TC ✓
```

```
C:\TC>TC ✓
```

（命令中的 ✓ 表示回车键）

Turbo C 的运行界面，如图 1-1 所示。屏幕的第一行为主菜单行，主菜单包括如下内容：File（文件）、Edit（编辑）、Run（运行）、Compile（编译）、Project（工程）、Option（选项）、Debug（调试）、Break/watch（断点及跟踪）。

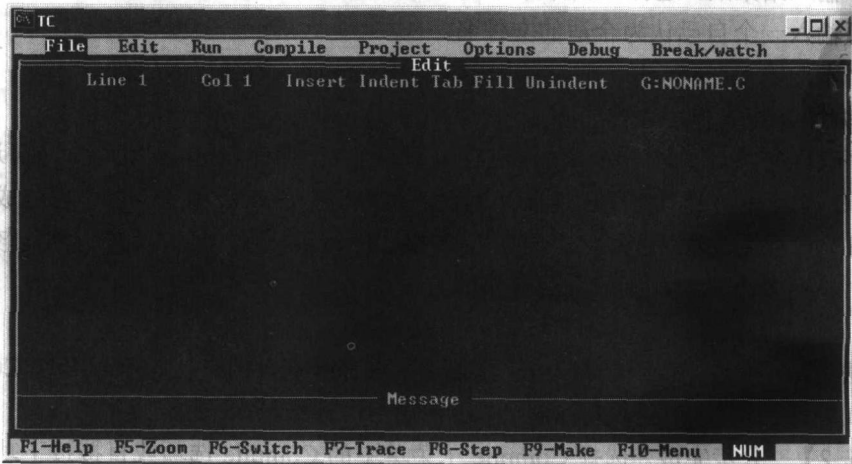


图 1-1 Turbo C 集成环境的屏幕界面



Edit 下为程序编辑区，用于输入、编辑 C 语言源程序。其中 Line 跟踪显示光标所在行；Col 跟踪显示光标所在列；Insert 表示当前输入为插入模式，用 Insert 键控制；Indent 为自动缩进模式，用 Ctrl+O+I 组合键控制；Tab 为制表符，用 Ctrl+O+T 组合键控制；C:Noname.c 位置显示当前所编辑的文件名，缺省文件名为 C:Noname.c。

Message 下为错误信息区，用于显示编译中发现的各种错误信息。最后一行为功能键提示行。

1.4.4 Turbo C 集成环境的使用

1. 菜单项的使用

主菜单中的每一项（除 Edit 之外）都有子菜单，按下 Alt 键的同时按下某个菜单项的第一个字母，即可下拉出相应选项的子菜单。例如，按下 Alt+F 即可进入 File 的下拉子菜单。

功能键 F10：是菜单行与程序编辑区的转换键。启动 Turbo C 后，程序光标停留在程序编辑区，此时，按下 F10 键，光标即可进入菜单行，利用左右箭头键可以选择相应的菜单选项。当光标在菜单行时，按下 F10 键光标即可返回程序编辑区。

功能键 F6：是程序编辑区与错误信息区的转换键，当光标停留在错误信息区时，按下 F6 键可以将光标切换到程序编辑区，反之亦可。

2. 编辑、编译及运行

下面举例说明 Turbo C 集成环境的使用。

（1）输入源程序。

进入 Turbo C 后，即可在程序编辑区输入 C 语言源程序。

例如，输入程序如下：

```
main( )
{
    printf("Turbo C集成环境的使用介绍!\n");
}
```

（2）存储源程序。

源程序输入结束后，选择 File 菜单中的 Save 选项或直接按下功能键 F2，系统将询问文件名，用户可输入一个自己认为合适的文件名。

例：本例输入 Example.c 后，源程序将以 Example.c 为名存入磁盘。

编译、连接和运行程序。

源程序存入磁盘后，即可对该程序进行编译、连接。编译、连接结束后，即可运行该程序。

Turbo C 集成环境中，编译、连接和运行，可以合并成一步完成，也可以分成几步分别完成，具体情况可视用户程序的大小而确定。如果程序较小，用一步完成的方法比较方便；如果程序很大，包括多个模块，则应采用分步的方法。

例如，本例程序很小，可以采用一步完成的方法。

【方法 1】 一步完成。选择 Run 子菜单中的 Run 选项或直接按下组合键 Ctrl+F9，系统将自动完成编译、连接和运行的全部工作，并输出运行结果。

【方法 2】 分步完成。选择 Compile 子菜单中的 Compile 选项或直接按下功能键 F9，系统只完成编译与连接工作，生成相应的 .EXE 文件。需要运行程序时，选择 Run 子菜单中的

Run 选项即可。也可以回到 DOS 状态下直接运行 .EXE 文件。

(4) 重新编辑。

如果编译或连接过程中发现程序错误,可按任意键回到编辑窗口,程序中的出错行将呈高亮度显示,同时在信息窗口出现一行或多行错误信息提示。使用箭头键(↑或↓)可以查看其他错误行。用户可以在编辑窗口中改正程序中的错误,然后重新编译、连接并运行。

(5) 查看结果:

程序的运行结果有两种查看方法。

【方法1】 按下 Alt+F5 键,可以查看程序的运行结果。

本例程序的运行结果如下:

Turbo C 集成环境的使用介绍!

查看结果后,按下任意键,仍可返回 Turbo C 环境。

【方法2】 运行结束后,退出 Turbo C 环境,即可看到程序的运行结果。

1.4.5 退出 Turbo C 环境

按下 Alt+X 键或选择 File 子菜单中的 Quit 选项,即可退出 Turbo C 集成环境,回到 DOS 状态。如果用户程序没有存盘,系统将提示用户存盘。

C 语言源程序被编译、连接后生成的 .exe 文件,任何时候都可以在 DOS 状态下运行。

1.4.6 其他常用菜单选项

1. 文件菜单 File

(1) Load: 装入 C 语言程序文件。从磁盘上调出一个 C 语言程序文件并装入内存,如果正在编辑的程序文件没有存盘,系统将提示用户存盘。

(2) New: 创建一个新文件。进入编辑窗口编辑新的文件,文件名默认为 Noname.c。

(3) Save: 保存当前编辑的 C 语言程序文件。将编辑窗口中的程序保存到磁盘上,如果此时文件名为 Noname.c,系统将提示用户输入新的文件名。

(4) Write to: 换名保存。将正在编辑的 C 语言程序文件以另外一个名字或路径保存到磁盘上。

(5) Directory: 显示目录。显示目录及所需要的文件,按下回车键进入当前目录。

(6) Change dir: 改变目录。显示当前目录,并允许用户将其改变为所指定的盘符和目录。

(7) Os shell: 进入 DOS 状态。暂时退出 Turbo C 环境并进入 DOS 状态,进行用户所需要的操作(例如拷贝、删除文件等 DOS 操作),键入 EXIT 返回 Turbo C 环境。

2. 编译菜单 Compile

(1) Compile to OBJ: 编译产生 .obj 文件。显示将产生的文件名,文件名的来源:初始的文件名 Noname.c 或编辑窗口的当前文件名。

(2) Make EXE file: 生成 .exe 文件。该菜单项是制作工程文件的命令。显示将产生的工程文件名,文件名的来源:在 Project 菜单的 Project name 子菜单项中指定的工程文件名、初始的文件名 Noname.c 或编辑窗口的当前文件名。

(3) Link EXE file: 连接并生成 .EXE 文件。连接当前的 .OBJ 文件和 .LIB 库文件,生