



高职高专 **精品课程** 规划教材 计算机系列

C 语言程序设计

向 华 主 编
杨 焰 杨 霞 姚 蕾 副主编



- 以国家级、省级优秀精品课程为基础
- 配有强大的网络教学资源：教学视频、案例、项目实践等
- 提供网上实践平台，可直接进行系统化、项目化实践
- 实现课程结构与内容实战化、职业化

清华大学出版社



高职高专精品课程规划教材 计算机系列

C 语言程序设计

向 华 主 编

杨 焰 杨 霞 姚 蕾 副主编

清华大学出版社

北 京

内 容 简 介

本书详细地介绍了C语言的基础知识,包括数据类型、结构化程序设计及相关控制语句、数组、函数、指针、结构体和共用体、位运算及文件等。

本书在编写上体现了任务驱动式教学思想,每一章的开头均围绕本章的学习目标提出了一个总体编程任务,该任务又分解为若干易完成的小任务,然后通过与任务相关知识的学习,逐步达到完成本章任务的目的。

本书强调实际编程能力的培养,知识结构完整、例题设计精心、习题丰富多样。除了每章末尾的上机实训之外,全书还根据大的教学环节设计了3个综合项目实训。在综合项目实训中,通过引导学生完成一个较复杂项目的设计、编程和调试,来培养及训练学生的程序设计技能以及分析问题和解决问题的能力。

本书中的示例代码均经过细心调试,保证能够正确运行。

本书既适于作为大专院校及高职高专相关专业的教材,又可作为成人教育和在职人员的培训教材,也可作为C语言编程爱好者的自学参考书。

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/向华主编;杨焰,杨霞,姚蕾副主编. —北京:清华大学出版社,2008.6

(高职高专精品课程规划教材 计算机系列)

ISBN 978-7-302-17689-3

I. C… II. ①向… ②杨… ③杨… ④姚… 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(2008)第074099号

责任编辑:石伟 宋延清

装帧设计:杨玉兰

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

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

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

印 刷 者:北京市世界知识印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:19 字 数:453千字

版 次:2008年6月第1版 印 次:2008年6月第1次印刷

印 数:1~4000

定 价:29.50元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:027821-01

《高职高专精品课程规划教材》丛书序

教育部于 2003 年开始在全国高等学校(包括高职高专院校)中启动高等学校教学质量与教学改革工程精品课程建设工作(简称精品课程建设),随后许多省份和高校也启动了省级和校级精品课程建设工作。经过 4 年的发展,精品课程建设已经进入成熟期,网上涌现了大量的优质课程资源,充分利用这些资源,无论对学生学习课程还是教师的教学都有积极的促进意义。

系列化的优秀教材与精品课程相呼应非常有必要,不但可以使优质的教学资源以教材为载体最大程度地得到共享和利用,而且教材的沉淀、积累和推广也将反过来促进精品课程资源的不断完善。

现在各个高职高专院校都以就业为导向,把对学生的技能培养作为首要目标。因此本套丛书以“体现职业教育教学特点和强调现代教育技术应用”为原则,以提高课堂与实践的教学效率和效果为主旨,努力建设一套全新的、有实用价值的精品课程配套规划系列教材,并希望能够通过这套教材的出版和使用,促进优秀精品课程的发展,最大限度地发挥精品课程的“精品”与“网络课程资源立体化”的优势,使之成为一套适应社会需求,有利于培养高素质高技能人才的优秀系列教材。

本系列丛书具有以下特点:

- 以国家级、省级优秀精品课程为基础。
- 配有强大的网络教学资源:教学视频、案例、项目实践等。
- 提供网上实践平台,可直接进行系统化、项目化实践。
- 实现课程结构与内容实战化、职业化。

精品课程在各个方面都已经比较成熟,所以本丛书力求在实用性上更加突出,注重技术能力的培养,提倡动手实践。每个单元小节后都有必要的习题和实训案例。大部分教材还专门配有实验与实训指导。使读者在掌握基本知识的同时,还可以获得实际操作的基本技能。

每本教材都配有内容细致全面的网站,为教师免费提供电子教案、案例库、习题库;为教师和学生共同开设专题讨论网络空间,可实现更大范围的教与学互动,即时解决教学过程中遇到的问题。在帮助老师教学的同时更能培养学生的学习兴趣,通过自己动手实践来提高专业技术能力。

本系列教材主要针对高职高专院校,以三年制高职教育为主,同时也适用于同等学历的职业教育。希望通过本系列教材的编写和推广应用,不仅能够有利于提高精品课程的整体水平,而且能够通过精品课程成熟的教学经验和丰富的网络教学资源更有助于加快职业技术教育的改革步伐。

精品课程网址为: <http://www.wenyuan.com.cn/jpkc>。

前 言

C 语言是比较流行的程序设计语言之一，特别适合作为初学者学习结构化程序设计的入门语言。本书作者主持的“C 语言程序设计”精品课程被评为 2007 年四川省精品课程，作者根据多年的 C 语言程序设计教学及教改经验编写了本书。

本书的主要特色如下：

- 体现了任务驱动式教学思想。每一章的开头均围绕该章的学习目标提出了一个总体编程任务，该任务又分解为若干易完成的小任务，然后通过对与任务相关的知识的学习，逐步达到完成该章任务的目的。
- 在例题的设计上结合了实用性和趣味性，既有利于启发思维，又能提高学生的学习积极性。
- 各章末尾均设有题型多样、题量丰富的习题，以及目的明确、内容详尽的上机实训，非常有利于教学。
- 重视实际编程能力的培养。除每章末尾安排了上机实训的内容外，全书还根据大的教学环节设计了 3 个综合项目实训。在综合项目实训中，通过引导学生完成一个较复杂项目的设计、编程和调试，来培养和训练学生的程序设计技能以及分析问题和解决问题的能力。

本书的第 1、6、7 章由成都职业技术学院的向华副教授编写，第 2、3、8 章由成都职业技术学院的杨霞老师编写，第 4、5 章由成都职业技术学院的姚蕾老师编写，第 9、10、11 章由成都职业技术学院的杨焰老师编写。

本书在编写过程中，得到了成都职业技术学院各级领导的大力支持。成都职业技术学院计算机系的周察金、卓国锋、汪剑、张渝、郭朗、李伟等老师对本书的编写给予了许多帮助，并为本书的校对做了大量的工作，在此一并表示衷心的感谢！

同时，也要感谢相关人员家人的支持和鼓励。

由于作者水平有限，书中难免有疏漏和不足之处，敬请读者批评指正。

目 录

第 1 章 C 语言概述	1	2.5.1 类型的自动转换	27
1.1 C 语言简史及特点	1	2.5.2 类型的强制转换	28
1.1.1 C 语言的发展	1	2.6 运算符与表达式	29
1.1.2 C 语言的特点	2	2.6.1 运算符简介	29
1.2 C 语言程序	2	2.6.2 算术运算符和算术表达式	30
1.2.1 几个典型的 C 程序	2	2.6.3 赋值运算符和赋值表达式	32
1.2.2 C 程序的基本结构	4	2.6.4 逗号运算符和逗号表达式	34
1.2.3 C 语言的基本符号与词汇	5	2.6.5 其他常用运算符	34
1.3 C 语言集成开发环境	6	2.6.6 运算符的优先级与结合性	35
1.3.1 Turbo C 集成开发环境介绍	6	2.7 上机实训：基本数据类型的	
1.3.2 Turbo C 集成开发环境		简单程序设计	36
的使用	7	2.7.1 实训目的	36
1.3.3 其他 C 语言集成开发		2.7.2 实训内容	36
环境简介	10	2.8 习题	38
1.4 上机实训：Turbo C 的基本操作	12	第 3 章 顺序结构程序设计	40
1.4.1 实训目的	12	3.1 算法	40
1.4.2 实训内容	12	3.1.1 算法的概念	40
1.5 习题	14	3.1.2 算法的表示	42
第 2 章 基本数据类型、运算符和		3.1.3 结构化程序设计方法	44
表达式	16	3.2 C 语句	45
2.1 基本数据类型	16	3.2.1 控制语句	45
2.1.1 数据类型	16	3.2.2 表达式语句	46
2.1.2 常量与变量	17	3.2.3 特殊语句	46
2.2 整型数据	19	3.3 数据输出	47
2.2.1 整型常量	19	3.3.1 输入/输出的概念	47
2.2.2 整型变量	21	3.3.2 格式输出函数 printf	47
2.3 实型数据	22	3.3.3 字符输出函数 putchar	51
2.3.1 实型常量	22	3.4 数据输入	52
2.3.2 实型变量	23	3.4.1 格式输入函数 scanf	52
2.4 字符型数据	24	3.4.2 字符输入函数 getchar	56
2.4.1 字符常量	24	3.5 顺序结构程序设计举例	57
2.4.2 字符变量	25	3.6 上机实训：顺序结构程序设计	58
2.4.3 字符串常量	26	3.6.1 实训目的	58
2.5 不同类型数据的混合运算	27	3.6.2 实训内容	59

3.7 习题	60	5.10 习题	110
第 4 章 选择结构程序设计	64	第 6 章 数组	113
4.1 关系运算符和关系表达式	64	6.1 一维数组的定义和引用	113
4.1.1 关系运算符	64	6.1.1 一维数组的定义	113
4.1.2 关系表达式	65	6.1.2 一维数组的引用	115
4.2 逻辑运算符和逻辑表达式	65	6.1.3 一维数组的初始化	118
4.2.1 逻辑运算符	65	6.2 二维数组的定义和引用	120
4.2.2 逻辑表达式	66	6.2.1 二维数组的定义	120
4.3 if 语句	68	6.2.2 二维数组的引用	122
4.3.1 最基本的 if 语句	68	6.2.3 二维数组的初始化	123
4.3.2 if else 语句	70	6.3 字符数组	125
4.3.3 多分支选择	72	6.3.1 字符数组的定义和初始化	125
4.4 switch 语句	75	6.3.2 字符数组的引用	126
4.5 上机实训: 选择结构程序设计	78	6.3.3 字符数组与字符串	127
4.5.1 实训目的	78	6.3.4 字符串的输入、输出和 处理函数	128
4.5.2 实训内容	79	6.4 上机实训一: 数值型数组	135
4.6 习题	82	6.4.1 实训目的	135
第 5 章 循环结构程序设计	85	6.4.2 实训内容	135
5.1 while 语句	85	6.5 上机实训二: 字符型数组	138
5.2 do-while 语句	87	6.5.1 实训目的	138
5.3 for 语句	89	6.5.2 实训内容	138
5.4 goto 语句	92	6.6 习题	140
5.5 几种循环控制语句的比较	94	第 7 章 函数	143
5.6 break 语句和 continue 语句	96	7.1 函数概述	143
5.6.1 break 语句	96	7.1.1 模块化程序思想	143
5.6.2 continue 语句	96	7.1.2 C 函数的分类	145
5.7 循环嵌套	98	7.2 函数的定义	146
5.7.1 循环嵌套的几种形式	98	7.2.1 函数定义的一般形式	146
5.7.2 循环嵌套应用举例	99	7.2.2 有关函数定义的几点说明	147
5.8 上机实训: 循环结构程序设计	102	7.3 函数参数及返回值	149
5.8.1 实训目的	102	7.3.1 函数参数	149
5.8.2 实训内容	103	7.3.2 函数的返回值	151
5.9 综合项目实训	107	7.4 函数的调用	152
5.9.1 实训内容	107	7.4.1 函数的语句调用	152
5.9.2 程序分析	107	7.4.2 函数表达式调用	153
5.9.3 部分源程序清单	108	7.4.3 函数的嵌套调用	156
5.9.4 实训报告	108		

7.4.4 函数的递归调用.....	158
7.5 数组作函数参数.....	161
7.5.1 数组元素作函数参数.....	161
7.5.2 数组名作函数参数.....	162
7.6 变量的作用域和生存期.....	164
7.6.1 变量的作用域.....	164
7.6.2 变量的生存期.....	167
7.7 函数的作用域.....	171
7.7.1 内部函数.....	171
7.7.2 外部函数.....	171
7.8 上机实训一：函数的定义和调用.....	176
7.8.1 实训目的.....	176
7.8.2 实训内容.....	176
7.9 上机实训二：局部变量和 全局变量.....	179
7.9.1 实训目的.....	179
7.9.2 实训内容.....	179
7.10 综合项目实训.....	181
7.10.1 实训内容.....	181
7.10.2 程序分析.....	182
7.10.3 部分源程序清单.....	183
7.10.4 实训报告.....	185
7.11 习题.....	186
第8章 指针.....	191
8.1 指针的概念.....	191
8.1.1 指针和指针变量.....	192
8.1.2 指针变量的定义.....	193
8.1.3 指针变量的操作.....	193
8.2 指针运算.....	196
8.2.1 指针的赋值运算.....	196
8.2.2 指针的加减运算.....	196
8.3 指针与数组.....	196
8.3.1 指向数组的指针.....	197
8.3.2 通过指针引用数组元素.....	197
8.4 指针与字符串.....	198
8.4.1 指向字符串的指针.....	198
8.4.2 字符串指针变量与字符数组 的区别.....	199

8.5 指针与函数.....	200
8.5.1 函数指针变量.....	200
8.5.2 指针型函数.....	201
8.6 指向指针的指针.....	203
8.7 上机实训：指针的应用.....	205
8.7.1 实训目的.....	205
8.7.2 实训内容.....	205
8.8 习题.....	206
第9章 结构体和共用体.....	209
9.1 结构体类型概述.....	210
9.1.1 结构体类型的特点.....	210
9.1.2 结构体类型的定义.....	210
9.2 结构体类型变量的定义和引用.....	211
9.2.1 结构体类型变量的定义.....	211
9.2.2 结构体类型变量的初始化 和引用.....	213
9.3 结构体数组.....	215
9.3.1 结构体数组的定义.....	215
9.3.2 结构体数组的初始化.....	216
9.4 指向结构体类型数据的指针.....	219
9.4.1 结构体指针变量的定义 和初始化.....	219
9.4.2 结构体指针的应用.....	220
9.5 结构体与函数.....	221
9.5.1 结构体变量作函数参数.....	221
9.5.2 结构体类型的函数.....	224
9.6 链表.....	224
9.6.1 链表的概念.....	224
9.6.2 链表的实现.....	224
9.6.3 动态链表.....	225
9.6.4 链表的操作.....	226
9.7 共用体.....	232
9.7.1 共用体的概念.....	232
9.7.2 共用体变量的定义和引用.....	232
9.8 上机实训一：结构体的基本应用.....	239
9.8.1 实训目的.....	239
9.8.2 实训内容.....	239
9.9 上机实训二：链表的应用.....	240

9.9.1 实训目的	240	11.3 文件的读写	261
9.9.2 实训内容	240	11.3.1 字符的输入和输出	261
9.10 上机实训三：共用体的应用	241	11.3.2 格式输入和输出	264
9.10.1 实训目的	241	11.3.3 字符串的输入和输出	266
9.10.2 实训内容	242	11.4 随机文件的读写	269
9.11 习题	243	11.4.1 文件的定位	269
第 10 章 位运算	246	11.4.2 fread 函数与 fwrite 函数	270
10.1 位逻辑运算	246	11.5 上机实训：文件的读写	276
10.1.1 按位与	246	11.5.1 实训目的	276
10.1.2 按位或	247	11.5.2 实训内容	276
10.1.3 按位异或	248	11.6 综合项目实训	279
10.1.4 按位取反	249	11.6.1 实训内容	279
10.2 移位运算	250	11.6.2 程序分析	279
10.2.1 左移位	250	11.6.3 部分源程序清单	280
10.2.2 右移位	250	11.6.4 实训报告	282
10.3 上机实训：位运算的应用	252	11.7 习题	283
10.3.1 实训目的	252	附录 1 常用 ASCII 码对照表	286
10.3.2 实训内容	252	附录 2 C 语言运算符的优先级和 结合性	287
10.4 习题	253	附录 3 Turbo C 常用库函数	288
第 11 章 文件	256	参考文献	291
11.1 C 语言文件概述	257		
11.2 文件的打开与关闭	258		

第 1 章 C 语言概述

C 语言是编程语言中较为流行的一种。随着计算机的普及和发展, 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。

本章内容

- C 语言的发展及特点。
- C 程序的基本结构。
- C 语言的基本符号与词汇。
- C 语言集成开发环境。

学习目标

- 掌握 C 程序的基本结构。
- 掌握 C 语言的基本符号与词汇。
- 掌握 Turbo C 集成开发环境的基本使用方法。
- 能够编写并在 Turbo C 中编辑和运行最简单的 C 程序。

本章任务

本章要完成的主要任务是编写并在 Turbo C 集成开发环境中运行一个简单的 C 程序, 该程序的功能是求两个整数之和。任务可以分解为两部分:

- 编写程序——了解 C 程序的基本结构。
- 学会在 Turbo C 集成开发环境中编辑并运行程序。

1.1 C 语言简史及特点

1.1.1 C 语言的发展

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, 并加进了多程序设计的功能, 称为 Unix 第五版, 开创了 Unix 系统发展的新局面。1975 年 Unix 第六版颁布后, C 语言得到计算机界的普遍认可, 从此, C 语言与 Unix 系统一起互相促进并获得迅速发展。

设计 C 语言的最初目的只是为了描述和实现 Unix 操作系统。而目前, 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:

- C 语言是结构化程序设计语言。C 语言程序的逻辑结构可以用顺序、选择和循环三种基本结构组成, 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。
- 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及软件工程技术的应用提供了强有力的支持。
- C 语言具有丰富的运算能力。C 语言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。
- C 语言具有丰富的数据类型和较强的数据处理能力。C 语言不但具有整型、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。
- 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。
- C 语言不但具有高级语言的特性, 还具有汇编语言的特点。C 语言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。
- C 语言具有较好的通用性。C 语言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 语言程序

1.2.1 几个典型的 C 程序

C 语言的源程序由一个或多个函数组成, 每个函数完成一种指定的操作, 所以有人又把 C 语言称为函数式语言。下面, 通过 3 个简单的例子来了解 C 程序的基本结构。

【例 1.1】在屏幕上显示信息“Welcome!”。程序代码如下:

```
main()
{
    printf("Welcome!\n");
}
```

运行结果:

Welcome!

程序说明:

- C 程序由一系列函数组成, 这些函数中必须有且只能有一个名为 `main` 的函数, 这个函数称为主函数, 整个程序从主函数开始执行。在例 1.1 中, 只有一个主函数而无其他函数。
- 程序第 1 行中的 `main` 是主函数的函数名, `main` 后面的一对小括号是函数定义的标志, 不能省略。
- 程序第 3 行的 `printf` 是 C 语言的格式输出函数, 在本程序中, `printf` 函数的作用是输出括号内双引号之间的字符串, 其中 “\n” 代表换行符。第 3 行末尾的分号则是 C 语句结束的标志。
- 程序第 2 行和第 4 行是一对大括弧, 在这里表示函数体的开始和结束。一个函数中要执行的语句都写在函数体中。

【例 1.2】 求一个整数的平方。程序代码如下:

```
main()
{
    int num, square;      /* 定义两个整型变量 */
    num = 9;              /* 把 9 赋值给变量 num */
    square = num * num;    /* 计算 num 的平方, 并将计算结果赋值给变量 square */
    printf("square=%d", square); /* 输出变量 square 的值 */
}
```

运行结果:

square=81

程序说明:

- 这个程序由一个主函数组成, 其中, 第 3 行的 `int` 表示定义变量类型为整型, 该行定义了 `num`、`square` 这两个整型变量。
- 程序第 4 行和第 5 行中的语句均为赋值语句, “=” 为赋值运算符, 其作用是将其右边的常量或表达式的值赋值给左边的变量。
- 第 6 行中的 “%d” 是输入输出函数中的格式字符串, 在此表示以十进制整数的形式输出变量 `square` 的值。程序的运行结果中, “%d” 的位置被 `square` 变量的值(即 81)取代。
- 程序中多次出现的 “/*” 和 “*/” 是一对注释符, 注释的内容写在这对注释符之间。注释内容对程序的编译和运行不起任何作用, 其目的是为了提高程序的可读性。在必要的地方给程序加上注释是一个好习惯, 这使得程序易于理解, 而对程序的理解是进一步修改和调试程序的基础。

【例 1.3】 输入两个整数, 输出其中的最大值。程序代码如下:

```
main()
{
    int a, b, m;          /* 定义 3 个整型变量 */
```

```

scanf("%d%d", &a, &b);      /* 输入两个整数到变量 a 和 b 中 */
m=max(a, b);    /* 调用 max 函数求 a 和 b 的最大值，并把函数的返回值赋值给变量 m */
                /* a 和 b 为实际参数 */
printf("max=%d", m);      /* 输出变量 m 的值 */
}

max(n1, n2)      /* 定义 max 函数，n1、n2 为形式参数 */
int n1, n2; /* 定义形式参数的类型。也可合并到函数头中：max(int n1, int n2) */
{
    int t;
    if (n1>n2)
        t = n1;    /* 比较 n1 和 n2 的大小，将最大值赋值给变量 t */
    else
        t = n2;
    return t;      /* 返回变量 t 的值 */
}

```

运行结果：

```

15 86 ✓
max=86

```

程序说明：

- 该程序由两个函数组成，一个是 main 函数，一个是 max 函数。max 函数的功能是求两个整数的最大值，而数据的输入和输出则在 main 函数中实现。
- main 函数和 max 函数的定义是相互独立的。
- main 函数的第 5 行调用 max 函数时，分别把实际参数 a 和 b 的值传递给形式参数 n1 和 n2，因此，调用 max 函数的结果是求得了 a 和 b 的最大值。

1.2.2 C 程序的基本结构

通过上面这 3 个例子，可以对 C 程序的基本结构归纳如下。

(1) C 语言程序由函数构成。函数是构成 C 程序的基本单位，即 C 程序由一个或多个函数组成，其中必须有且只能有一个名为 main 的主函数。例如，在前面的例 1.1 和例 1.2 中，均只有一个 main 函数，而在例 1.3 中，则有 main 和 max 两个函数。

(2) 每个函数的基本结构如下：

```

函数名()
{
    语句 1;
    ...
    语句 n;
}

```

有的函数在定义时，函数名后的小括号内有形式参数，如例 1.3 中的 max 函数。{} 内则是由若干语句组成的函数体，每个语句必须以分号结束。C 语言的书写格式较自由，一行内可以写多个语句，一个语句很长时也可以分写在多行上。

(3) 各个函数的定义是相互独立的。各函数定义的顺序无关紧要，主函数可以定义在

其他函数之前，也可以定义在其他函数之后，但程序的执行总是从主函数开始。

1.2.3 C语言的基本符号与词汇

任何程序设计语言都规定了自己的一套基本符号和词汇，C语言也不例外。

1. C语言的基本符号集

C语言的基本符号集采用ASCII码字符集，包括：

- 大小写英文字母各26个。
- 10个阿拉伯数字0~9。
- 其他特殊符号，包括以下运算符和操作符：

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

2. C语言的词汇

(1) 标识符

程序中用来标识变量名、函数名、数组名、数据类型名等的有效字符序列称为标识符。简单地说，标识符就是一个名字。

标识符的构成规则如下。

① 标识符只能由英文字母(A~Z、a~z)、数字(0~9)和下划线(_)三类符号组成，但第一个字符必须是字母或下划线。

例如，下面的标识符是合法的：

sum、Sum、n2、_average、a_3、student_2_name

下面是不合法的标识符：

num-1、a#3、2student、!sum_2、number.3

② 大写字母与小写字母含义不同(即C语言是对大小写敏感的)。如sum、Sum、SUM表示三个完成不同的标识符。

③ 一般的C编译系统只取标识符的前8个字符为有效字符，而Turbo C则取标识符的前32个字符为有效字符。

④ 通常，命名标识符时应该做到“常用取简，专用取繁”。一般情况下，标识符的长度在8个字符以内就可以了。

(2) 关键字

关键字又称为保留字，是C语言编译系统所固有的、具有专门意义的标识符。C语言的关键字有32个，一般用作C语言的数据类型名或语句名，如表1.1所示。

表 1.1 C 语言关键字

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

说明：

- ① 所有关键字的字母均采用小写。
- ② 关键字不能再作为用户的常量、变量、函数和类型等的名字。

 在学习了上述相关知识之后，我们通过下例来完成本章开篇提出的任务之一。

【例 1.4】输入两个整数，输出这两个整数之和。程序代码如下：

```
main()
{
    int a, b, sum;           /* 定义 3 个变量 */
    scanf("%d%d", &a, &b);   /* 输入两个整数到变量 a 和 b 中 */
    sum = a + b;             /* 求 a 和 b 的和，并把结果放入变量 sum 中 */
    printf("sum=%d\n", sum); /* 输出变量 sum 的值 */
}
```

运行结果：

```
35 29 ✓
sum=64
```

1.3 C 语言集成开发环境

1.3.1 Turbo C 集成开发环境介绍

C 语言有许多种编译器，这些编译器之间只有很小的差别，只要学会其中的一种，对其他几种就能很快地适应。本节主要介绍在微机上较为流行的 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 具有图形库和函数库，具有查错及协处理器的仿真功能。当前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 操作系统下运行。本书的所有例题都在该环境下调试通过，后面的介绍都以此为基础。

1.3.2 Turbo C 集成开发环境的使用

1. 启动 Turbo C

假设 Turbo C 安装在 C 盘根目录下的 TC 文件夹中，那么在 Windows XP 下运行 Turbo C 的步骤如下。

(1) 单击屏幕左下角的“开始”按钮，选择“所有程序”→“附件”→“命令提示符”，即可打开“命令提示符”窗口。也可以单击“开始”按钮，选择“运行”，再在弹出的“运行”对话框中输入“cmd”后，单击“确定”按钮，结果如图 1.1 所示。

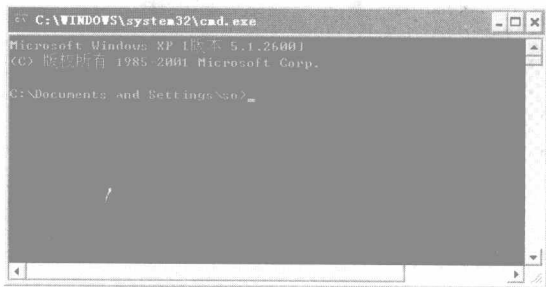


图 1.1 “命令提示符”窗口

(2) 在提示符后输入“cd\”，按 Enter 键后回到 C 盘根目录下。

(3) 在提示符后依次输入以下命令即可启动 Turbo C：

C:\>cd tc ✓

C:\TC>tc ✓

Turbo C 启动后，屏幕上即显示出其开始界面，如图 1.2 所示。

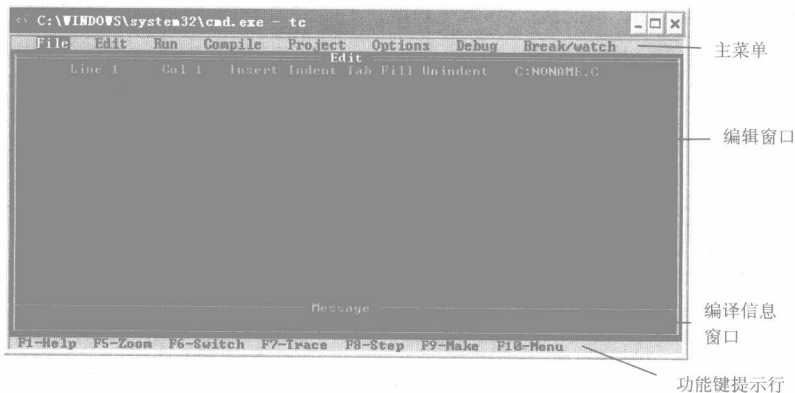


图 1.2 Turbo C 2.0 的开始界面

2. C 源程序的建立和编辑

(1) 源程序的建立

在 Turbo C 中新建一个 C 源程序时,可以直接选择 Edit(编辑)菜单,当光标停留在编辑窗口中时,就可以输入程序了。每输完一行后,按 Enter 键将光标移到下一行的开始位置。这里,在编辑窗口中输入例 1.1 中的程序,如图 1.3 所示。



图 1.3 输入例 1.1 中的程序

(2) 源程序的编辑

在编辑窗口中,可以很方便地对源程序作修改。表 1.2 列出了 Turbo C 的常用编辑命令。

表 1.2 Turbo C 的常用编辑命令

功能键	功 能	功能键	功 能
↑	光标上移一行	Delete	删除光标处的字符
↓	光标下移一行	Backspace	删除光标处的前一个字符
←	光标左移一个字符	Ctrl+Y	删除光标所在行
→	光标右移一个字符	Ins	插入、改写状态转换开关
PgUp	光标上移一页	Ctrl+K+B	定义块首
PgDn	光标下移一页	Ctrl+K+K	定义块尾
Home	光标移到当前行首	Ctrl+K+V	移动定义的块到光标所在处
End	光标移到当前行尾	Ctrl+K+C	复制定义的块到光标所在处
Ctrl+Home	光标移到源程序首	Ctrl+K+Y	删除定义的块
Ctrl+End	光标移到源程序尾	Ctrl+K+H	取消块的定义

3. 保存 C 程序

选择 File→Save(文件→保存)菜单命令,或直接按 F2 键,即可保存当前编辑窗口中的 C 程序,保存之前要求先给当前默认文件名为“NONAME.C”的 C 程序重命名。