

21世纪

高职高专创新精品规划教材

C 语言程序设计

主 编 朱建芳 周建辉

副主编 杜 娟 石明翔

创新
精品

“教、学、做”一体化，强化能力培养
“工学结合”原则，提高社会实践能力
“案例教学”方法，增强可读性和可操作性



中国水利水电出版社
www.waterpub.com.cn

21 世纪高职高专创新精品规划教材

C 语言程序设计

主 编 朱建芳 周建辉

副主编 杜 娟 石明翔



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书是“21 世纪高职高专创新精品规划教材”之一。

本书介绍 C 语言语法知识、程序设计基本方法、数据结构与算法基础知识,以及如何利用 C 语言编程解决问题和进行应用项目开发。主要内容有: C 语言概述、C 语言集成开发环境 Turbo C 的操作使用、C 语言语法知识、C 语言的数据输入输出处理、3 种基本结构(顺序、选择、循环)的 C 语言程序设计、程序设计基本方法、C 语言模块化程序设计、数据结构与算法的基础知识、C 语言的综合应用、实训项目等。每章后面均配有丰富的习题和上机实践内容。

本书可用作高职高专院校计算机类专业和其他工科类相关专业 C 语言程序设计课程的教材,也可供其他 C 语言初学者参考使用。

本书配有免费电子教案,读者可以从中国水利水电出版社网站以及万水书苑下载,网址为: <http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

C语言程序设计 / 朱建芳, 周建辉主编. -- 北京 :
中国水利水电出版社, 2010.1

21世纪高职高专创新精品规划教材
ISBN 978-7-5084-6960-7

I. ①C… II. ①朱… ②周… III. ①
C语言—程序设计—高等学校: 技术学校—教材 IV.
①TP312

中国版本图书馆CIP数据核字(2009)第205365号

策划编辑: 杨庆川 责任编辑: 宋俊娥 加工编辑: 庞永江 封面设计: 李 佳

书 名	21 世纪高职高专创新精品规划教材 C 语言程序设计
作 者	主 编 朱建芳 周建辉 副主编 杜 娟 石明翔
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 21 印张 531 千字
版 次	2010 年 1 月第 1 版 2010 年 1 月第 1 次印刷
印 数	0001—4000 册
定 价	35.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

序

近年来,我国高等职业教育蓬勃发展,为现代化建设培养了大量高素质技能型专门人才,对高等教育大众化作出了重要贡献,顺应了人民群众接受高等教育的强烈需求。高等职业教育作为高等教育发展中的一个类型,肩负着培养面向生产、建设、服务和管理第一线需要的高技能人才的使命,在我国加快推进社会主义现代化建设进程中具有不可替代的作用。随着我国走新型工业化道路、建设社会主义新农村和创新型国家对高技能人才要求的不断提高,高等职业教育既面临着极好的发展机遇,也面临着严峻的挑战。

教材建设是整个高职高专院校教育教学工作的重要组成部分,高质量的教材是培养高质量人才的基本保证,高职高专教材作为体现高职高专教育特色的知识载体和教学的基本工具,直接关系到高职高专教育能否为一线岗位培养符合要求的高技术性人才。中国水利水电出版社本着为高校教育服务,为师生提供高品质教材的原则,按照教育部《关于全面提高高等职业教育教学质量的若干意见》的要求,在全国数百所高职高专院校中遴选了一批具有丰富的教学经验、较高的工程实践能力的学科带头人和骨干教师,成立了高职高专教材建设编委会。编委会成员经过几个月的广泛调研,了解各高职院校教学改革和企业对人才需求的情况,探讨、研究课程体系建设和课程设置,达成共识,组织编写了本套“21世纪高职高专创新精品规划教材”。

本套教材的特点如下:

1. 面向高职高专教育,将专业培养目标分解落实于各门课程的技术应用能力要求,建立课程的技术、技能体系,将理论知识贯穿于其中,并融“教、学、做”为一体,强化学生的能力培养。

2. 理论知识的讲解以基础知识和基本理论“必需、够用”为原则,在保证达到高等教育水平的基础上,注重基本概念和基本方法讲解的科学性、准确性和正确性,把重点放在概念、方法和结论的阐释和实际应用上,推导过程力求简洁明了。

3. 在教材中按照技术、技能要求的难易和熟练程度,选择恰当的训练形式和内容,形成训练体系;确定实训项目,并将实训内容体现在教材中。对于单独设置实训的课程,我们将实训分成基础实训和综合实训两个部分。综合实训中重点体现了工学结合的原则,提高学生的社会实践能力。

4. 在编写方式上引入案例教学和启发式教学方法,采用以实际应用引出的问题为背景来设计和组织内容,增强了教材的可读性和可操作性,激发学生的学习兴趣,使知识点更容易理解掌握,从而使学生能够真正地掌握相关技术,为以后的就业打好基础。

5. 教材内容力求体现经济社会发展对应用技术的新要求和新趋势,将新兴的高新技术、复合技术等引进教材,并在教材中提出了一些引导技术发展的新问题,以期引起思考和讨论,有利于培养学生技术应用中的创新精神和能力。

6. 大部分教材都配有电子教案和相关教学资源,以使教材向多元化、多媒体化发展,满足广大教师教学工作的需要。电子教案使用 PowerPoint 制作,教师可根据授课情况任意修改。相关教案和资源可以从中国水利水电出版社网站 www.waterpub.com.cn 下载。

本套教材凝聚了众多奋斗在高等职业教育教学、科研第一线的教师和科研人员多年的教学经验和智慧，教材内容选取新颖、实用，层次清晰，结构合理，概念清晰，通俗易懂，可读性和实用性强。本套教材适用于高职高专院校，也可作为社会各类培训班用书和自学参考用书。

我们期待广大读者对本套教材提出宝贵意见和建议，以便进一步修订，使该套教材不断完善。

21 世纪高职高专创新精品规划教材编委会

2008 年 4 月

前 言

C 语言是应用最广泛的计算机程序设计语言之一, C 语言程序设计已成为当前高校几乎所有的计算机类专业及许多的理工类专业都开设的一门程序设计语言基础课程。针对 C 语言教学的教材已是琳琅满目, 但是现有的高职高专类 C 语言程序设计教材, 教学内容的选取和排序绝大多数还是基于“学科体系”的课程观, 以理论知识讲解为主, 技能训练为辅, 过于集中讲“是什么, 为什么”的概念、原理和理解方面的陈述性知识, 关于“如何做, 如何做得更好”的经验和策略方面的过程性知识讲得不够。本教材遵照“以就业为导向”、“以能力为本位”的高等职业教育指导思想的要求, 教学内容的选取和组织兼顾基本理论知识的学习和程序设计技能的训练, 做到陈述性知识和过程性知识的有机融合。在介绍陈述性知识(C 语言语法知识、数据结构与算法知识、程序设计方法等)的同时, 强调过程性知识(如何用 C 语言编程解决问题, 用哪一种编程方法效率更高、效果更好等)的获取。具有以下几个特点:

1. 作为一门专业基础课程, 基础理论知识内容的排序仍然遵循“学科体系”的原则, 由词法、句法到章法, 由浅入深, 由简单到复杂, 由局部到系统, 有别于专业技能课程内容的“行动体系”排序原则。

2. 技能训练内容分两部分。一是每章后的实训。在实训过程中创设疑难情境, 引导思考, 提出解决问题或关于知识难点的种种假设, 调动学生学习兴趣, 让学生在完成任务的情境中掌握知识和技能。二是综合应用实训项目。以一个电子通讯录的程序设计与开发为案例, 让学生在解决实际问题的过程中巩固知识及提高应用知识解决问题的能力。

3. 考虑到高职高专的教育对象倾向于形象思维的特点, 做到举例以通俗易懂、贴近生活和富有趣味性的内容为主, 淡化在复杂的数值计算方面的应用; 在容易出错的地方, 给出“注意”提示; 在重要知识点后给出“思考”提问, 以达到开拓思维、举一反三和融会贯通的效果; 在应用编程实例中给出“提示”, 指引读者分析问题、组织数据、设计算法和编写程序以及运行测试等。

本书共分 13 章, 主要内容包括: C 语言概述, 基本数据类型、运算符与表达式, 简单的 C 程序设计, 选择结构的 C 程序设计, 循环结构的 C 程序设计, 数组和串, 函数, 编译预处理, 指针, 结构类型, C 语言的模块化程序设计, 文件, C 语言的综合应用。最后提供一个电子通讯录的 C 语言程序设计与实现的完整过程作为实训项目参考。建议教学课时数为 72~90。

本书由广州航海高等专科学校的朱建芳副教授和北京现代职业技术学院的周建辉任主编, 杜娟和石明翔任副主编。第 1、3、4、5、8、11、13 由朱建芳编写, 第 7、9 章由周建辉编写, 第 2、6 章由杜娟编写, 第 10、12 章和实训项目由石明翔编写。第 2、6、7、9、10、12 章由周建辉审稿, 全书由朱建芳负责统稿和定稿。参加本书部分内容编写和程序调试的人员还有尹冬梅、李莉、傅宜宁和陈钟等。

由于作者水平有限, 书中难免有错漏和不妥之处, 恳请读者和专家批评指正。

编 者

2009 年 12 月

目 录

序	
前言	
第1章 C语言概述	1
本章学习目标	1
1.1 C语言的发展简史	1
1.2 C语言的特点	2
1.3 认识C程序	3
1.3.1 一个简单的C程序	3
1.3.2 C程序的基本结构	4
1.3.3 C程序中的单词	6
1.3.4 如何编写可读性良好的C程序	7
1.4 C程序的运行与调试	9
1.4.1 C程序设计的基本步骤	9
1.4.2 如何使用Turbo C/C++3.0运行 与调试C程序	12
习题一	15
实训一	16
第2章 基本数据类型、运算符与表达式	18
本章学习目标	18
2.1 C语言的数据类型	18
2.2 常量与变量	19
2.2.1 常量	19
2.2.2 变量	19
2.3 整型数据	20
2.3.1 整型常量	20
2.3.2 整型变量	21
2.4 实型数据	24
2.4.1 实型常量	24
2.4.2 实型变量	25
2.5 字符型数据	26
2.5.1 字符常量	26
2.5.2 字符变量	28
2.5.3 字符串常量	29
2.6 各类数据混合运算	30
2.7 C语言的运算符与表达式	31
2.7.1 算术运算符和算术表达式	32
2.7.2 自增与自减运算符	33
2.7.3 赋值运算符和赋值表达式	35
2.7.4 关系运算符及其表达式	38
2.7.5 逻辑运算符及其表达式	39
2.7.6 逗号运算符(,)及其表达式	42
2.7.7 位运算符及其表达式	43
2.8 运算符的优先级及其结合性	45
习题二	46
实训二	47
第3章 简单的C程序设计	49
本章学习目标	49
3.1 C语句概述	49
3.2 程序的三种基本结构	50
3.3 赋值语句	52
3.4 数据输入输出	52
3.4.1 printf()——格式化输出函数	53
3.4.2 putchar()——单个字符的输出函数	57
3.4.3 scanf()——格式化输入函数	57
3.4.4 getchar()——单个字符输入函数	60
3.5 顺序结构的简单C程序设计举例	60
习题三	62
实训三	65
第4章 选择结构程序设计	67
本章学习目标	67
4.1 if语句	67
4.2 switch语句	74
4.3 选择结构程序设计举例	76
习题四	79
实训四	83
第5章 循环结构的C程序设计	84
本章学习目标	84
5.1 循环语句	84

5.1.1 goto 语句	84	8.3 条件编译	159
5.1.2 while 语句	86	习题八	160
5.1.3 do-while 语句	88	实训八	161
5.1.4 for 语句	89	第 9 章 指针	163
5.2 break 语句和 continue 语句	91	本章学习目标	163
5.2.1 break 语句	91	9.1 指针与内存地址	163
5.2.2 continue 语句	92	9.1.1 变量的地址	163
5.2.3 break 语句和 continue 语句的比较	93	9.1.2 变量在内存中的存取	164
5.3 循环的嵌套	93	9.2 变量的指针与指针变量	165
5.4 几种循环的比较	97	9.2.1 什么是指针变量	165
5.5 循环结构的 C 程序设计举例	98	9.2.2 指针变量的定义	165
习题五	100	9.2.3 指针变量的引用	166
实训五	103	9.2.4 指针变量作为函数参数	169
第 6 章 数组和串	105	9.3 数组指针	172
本章学习目标	105	9.3.1 指向数组元素的指针	172
6.1 一维数组的定义和引用	105	9.3.2 通过指针引用数组元素	173
6.2 二维数组的定义和引用	110	9.3.3 数组名作函数参数	175
6.3 字符数组与字符串	116	9.3.4 二维数组的指针和指向二维数组 的指针变量	178
6.4 常用的字符串处理函数	119	9.4 字符串指针	181
6.5 数组应用编程	122	9.4.1 字符串的指针表示和引用	181
习题六	125	9.4.2 字符串指针作函数参数	183
实训六	127	9.5 函数指针	185
第 7 章 函数	130	9.5.1 指向函数的指针变量	185
本章学习目标	130	9.5.2 用函数指针变量调用函数	186
7.1 函数的定义	130	9.5.3 用指向函数的指针作函数参数	187
7.2 函数的调用	134	9.6 返回指针值的函数	189
7.2.1 函数调用的一般形式	134	9.7 指针数组和指向指针的指针	191
7.2.2 函数间的参数传递	136	9.7.1 指针数组	191
7.2.3 函数调用的方式	139	9.7.2 指向指针的指针	193
7.2.4 函数的声明	140	9.7.3 指针数组作 main() 函数的形参	195
7.2.5 函数的递归调用	141	9.8 指针应用编程	196
7.2.6 数组作为函数的参数	147	习题九	199
7.3 由多个函数组成的 C 程序设计	149	实训九	202
习题七	151	第 10 章 结构类型	205
实训七	153	本章学习目标	205
第 8 章 编译预处理	155	10.1 结构体类型与结构体变量的定义	205
本章学习目标	155	10.1.1 结构体类型的定义	205
8.1 宏定义	155	10.1.2 结构体类型变量的定义	206
8.2 文件包含	158		

10.2 结构体变量的引用与初始化	208	12.1.2 文件的操作流程	250
10.2.1 结构体类型变量的引用	208	12.1.3 文件指针	251
10.2.2 结构体变量的初始化	209	12.2 文件的打开与关闭	251
10.3 结构体数组	210	12.2.1 文件的打开	251
10.3.1 结构体数组的定义	210	12.2.2 文件的关闭	253
10.3.2 结构体数组的初始化	211	12.3 文件的读写	253
10.3.3 结构体数组应用举例	211	12.3.1 读/写一个字符	254
10.4 指向结构体类型数据的指针	213	12.3.2 读/写一个字符串	256
10.4.1 指向结构体变量的指针	213	12.3.3 读/写一个数据块	257
10.4.2 指向结构体数组的指针	214	12.4 文件定位	259
10.5 共用体类型	215	12.4.1 rewind()和 fseek()函数	260
10.5.1 共用体类型的定义	215	12.4.2 返回文件当前位置 ftell()函数	261
10.5.2 共用体变量的定义	216	12.5 文件的出错检测	262
10.5.3 共用体变量的引用	217	12.5.1 ferror()函数	262
10.6 枚举类型	219	12.5.2 clearerr()函数	262
10.7 用 typedef 定义类型的别名	220	习题十二	262
10.8 结构体应用编程	221	实训十二	264
习题十	224	第 13 章 C 语言的综合应用	266
实训十	227	本章学习目标	266
第 11 章 C 语言的模块化程序设计	230	13.1 数据结构与算法的概念	266
本章学习目标	230	13.1.1 数据结构	266
11.1 程序设计的基本方法	230	13.1.2 算法	270
11.1.1 自顶向下、逐步求精的设计过程	230	13.2 顺序表的插入与删除	273
11.1.2 模块化程序设计	233	13.2.1 什么是顺序表	273
11.1.3 结构化编程	234	13.2.2 顺序表的插入和删除	274
11.2 C 语言对模块化编程的支持	234	13.3 链表的插入与删除	278
11.2.1 局部变量与全局变量	234	13.3.1 什么是链表	278
11.2.2 变量的存储类型	237	13.3.2 单链表的建立、插入与删除	279
11.2.3 内部函数与外部函数	241	13.4 综合应用案例分析	284
11.3 多个源程序文件组成的 C 程序设计	243	习题十三	297
习题十一	245	实训十三	299
实训十一	248	实训项目 电子通讯录的实现	300
第 12 章 文件	249	附录 A ASCII 码表	321
本章学习目标	249	附录 B C 语言的词法、句法和章法提要	322
12.1 C 语言文件概述	249	附录 C Turbo C 常用库函数	323
12.1.1 文件的概念及分类	249	参考文献	327

第1章

C 语言概述

本章学习目标

本章主要介绍 C 语言的发展简史、C 语言的特点、C 程序的基本结构和组成要素以及 C 程序的运行与调试等内容。通过本章的学习，要求达到以下目的：

- 了解 C 语言的发展简史和特点
- 掌握 C 语言的基本结构及如何编写可读性良好的 C 程序
- 掌握一个 C 程序的编辑、编译、运行和调试的一般过程

1.1 C 语言的发展简史

C 语言是一种应用广泛的编程语言。用 C 语言不仅可以编写系统软件，如 UNIX、Microsoft Windows、dBASE 以及当今应用越来越广泛的嵌入式系统软件等，还可以编写出各种应用软件，如游戏软件、商业软件、工具软件、文字处理软件、图表软件、数学运算软件等。

任何一种计算机语言都有其发展的目的和特点。Basic 语言的主要目的是让初学者容易编写程序，特点是语法近似英文，浅显易懂；FORTRAN 语言是为了解决科学计算问题而产生；而 COBOL 语言则是为了解决商业和管理上的数据处理问题而发展起来的。还有，PASCAL 语言是第一个结构化语言。但是这些语言都因其发展背景以及语言本身的限制而无法兼顾实用和功能。C 语言则继承和发扬了许多高级语言的成功经验，同时弥补了它们的缺憾。

C 语言发源于著名的美国贝尔实验室（Bell Laboratory），是由该实验室的研究人员 Dennis Ritchie 和 Ken Thompson 两人于 20 世纪 70 年代初在设计 UNIX 操作系统时开发出来的。

C 语言发明前的操作系统等系统软件主要是用汇编语言编写的，原来的 UNIX 操作系统就是 1969 年由美国贝尔实验室的 Ken Thompson 和 Dennis Ritchie 利用汇编语言开发成功的。但是由于汇编语言依赖于计算机硬件，程序的可读性和可移植性都比较差。为了解决这个问题，贝尔实验室的 Ken Thompson 于 1970 年以 BCPL（英国剑桥大学的 Martin Richards 设计的比较接近硬件的一种高级语言）语言为基础，设计出一种简单而且接近硬件的高级语言——B 语言，并用 B 语言写了 UNIX 操作系统。

由于 B 语言是无类型的语言，过于简单，功能有限。随着对软件任务要求的日益提高，迫切需要设计一种更好、功能更强的语言。1972 年到 1973 年间，Dennis Ritchie 在 B 语言基础上设计了 C 语言，既保留了 B 语言的精炼和接近硬件的优点，又增加了函数、整型数、浮点数、字符、几十种操作符和其他成分。后来，C 语言又经过多次改进，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》，大大提高了 C 语言的可移植性，

同时也推动了 UNIX 操作系统迅速在各种机器上的实现。到 1978 年, C 语言已经发展成为最流行的几种计算机编程语言之一。

随着 C 语言的使用日益广泛, 出现了各种 C 编译系统, 为了使 C 语言更加规范化, 1983 年美国国家标准化协会 (ANSI) 开始对 C 的发展和扩充制定了新的标准, 称作 ANSI C。1987 年又公布了新标准——87 ANSI C, 该标准于 1990 年被国际标准化组织 (ISO) 接受为国际标准。Microsoft C、Turbo C、Quick C、Borland C 等流行的 C 编译系统都是以 ANSI C 为基础, 在此之上不同版本略有不同。本书的叙述以 ANSI C 为基础, 以 Turbo C/C++3.0 为编译环境, 所有范例程序均在 Turbo C/C++3.0 环境调试通过。

C 语言的发展一直没有结束。20 世纪 80 年代后, 许多著名的软件公司都对 C 语言的功能做了大量的扩充, 先后出现了 C++、Visual C++、C#, 使 C 语言继续向面向对象、可视化和网络化的程序设计语言发展。

C 语言已代替过去的 Pascal 语言成为高等院校计算机专业以及其他一些工科类专业学生必修的一门计算机语言课程。现在计算机专业学生必修的“数据结构”以及“操作系统”课程基本上都以 C 语言作为先修课程, 因为这两门课程中的算法描述和程序设计大多使用 C 语言。所以学好 C 语言非常必要, 而且掌握了 C 语言之后, 就可以毫不困难地学习并掌握其他程序设计语言。因此, C 语言值得你投资时间和精力去学习与研究。

1.2 C 语言的特点

C 语言之所以能够发展成为最受欢迎和最优秀的语言之一, 是因为 C 语言具有如下几个方面的特点:

(1) 语言简洁、灵活, 程序执行效率高。C 语言只有 32 个关键字, 对数据的描述简单, 各种语句的形式简单, 程序书写自由, 源程序的编写方便。用 C 语言编写的程序比用其他高级语言编写的程序执行效率高, 只比用汇编语言编写的代码低 10%~20%。

(2) 运算符丰富。C 语言共有 34 种运算符, 运算类型极为丰富, 表达式类型多样化, 能实现在其他高级语言中难以实现的运算功能。

(3) 数据类型丰富。C 语言具有整型、实型、字符型、数组、指针、结构体、共用体、枚举等类型, 可实现各种复杂的数据结构的运算。

(4) 灵活的结构化控制语句。C 语言具有条件选择、多分支选择、循环等多种控制语句, 是理想的结构化程序设计语言, 而且 C 语言以函数为程序基本单位, 容易实现模块化程序设计。

(5) 语法不严, 程序设计自由。C 语言对语法检查相对较宽松。如, 对变量类型使用比较灵活, 如整型数据与字符型数据可以通用; 逻辑表达式中参与运算的量值为非 0 即代表“真”, 而不像其他语言只有值为 1 才代表“真”, 等等。语法限制不严, 让编程人员可以更加灵活地进行程序设计。

(6) 具有直接对硬件进行控制的功能。C 语言能够通过位运算直接对硬件进行操作, 可以实现汇编语言的大部分功能, 这是 C 语言有别于其他高级语言的最突出的一大特点, 也是 C 语言可用来编写系统软件的原因。所以 C 语言具有“高级语言”和“低级语言”的双重特性, 被称作“中级语言”。

(7) 可移植性好。C 语言和汇编语言相比, 不仅具有很强的硬件控制能力, 同时具有更

强的表达和运算能力,并且,C语言本身能够独立于硬件,从而具有很好的可移植性,所以很多大型的系统软件选用C语言编写。

(8) C语言的另一面。正是由于C语言语法限制不严、编程灵活,所以要求C语言程序员必须花更多的时间去学习、熟悉C语言语法。尤其是指针的应用,常常使初学者难以理解,从而产生C语言难学的印象。其实,这只是C语言给你的难以接近的第一印象,一旦你跟它熟悉后,你就会逐渐体会到它的魅力无穷并深深地喜欢上它。

上面列出的C语言的这些特点,读者恐怕一时难以理解,有待读者在以后的学习与应用过程中逐步领会。

1.3 认识C程序

1.3.1 一个简单的C程序

下面从一个简单的C程序开始进入C的世界。可以在任何喜爱的文本编辑器中编写C程序,编写完后再移植到C的编译器中编译执行。C的编译系统一般都提供相应的程序编辑器,如Turbo C/C++3.0、Microsoft C和Microsoft Visual C++等,所以一般都在C的编辑器中编写程序。下面就让我们在Turbo C/C++3.0的环境里编写第一个C语言程序。(关于Turbo C/C++3.0的详细操作说明请看1.4.2节。)

(1) 打开Turbo C/C++3.0环境,在编辑窗口输入下面的程序。

例 1-1 在屏幕显示字符串“Welcome to the C world!”。

```
/*exam1-1*/  
#include <stdio.h>  
main()  
{  
    printf("Welcome to the C world!\n");  
}
```

(2) 编写完程序后,对程序进行编译和连接,生成可执行文件。图1-1是编译、连接后的情形。

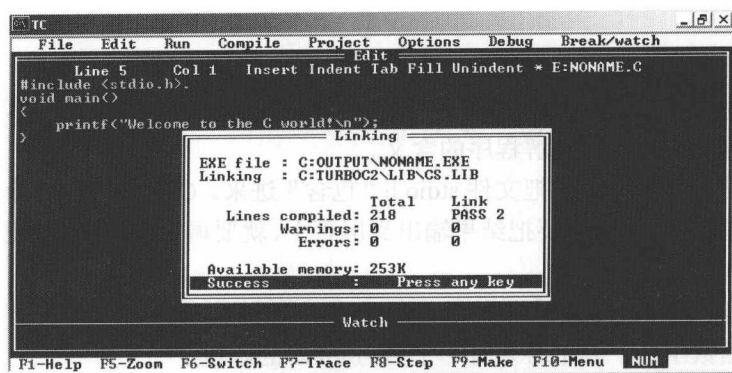


图 1-1 TC 的编译、连接提示界面

(3) 执行程序。转到用户输出窗口即可看到该程序的运行结果，如图 1-2 所示。

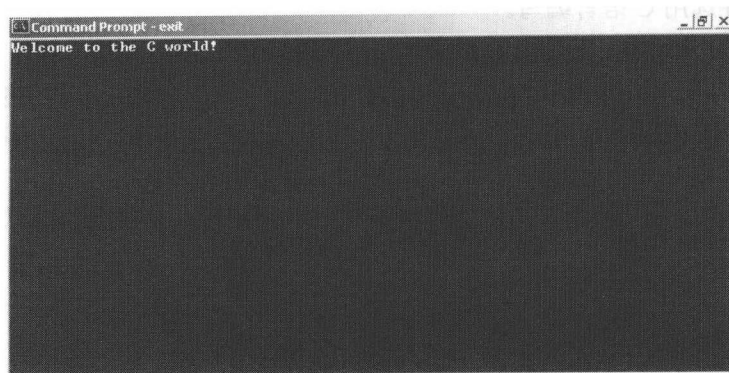


图 1-2 TC 的运行结果输出窗口

图 1-2 是程序运行后的用户输出窗口。

程序运行结果：

Welcome to the C world!

1.3.2 C 程序的基本结构

下面再来看另一个简单的 C 程序，并从这个例子出发说明 C 程序的基本结构。

例 1-2 计算两个数的和并输出结果。

```
/*exam1-2*/
#include <stdio.h>           /*将 stdio.h 这个文件包含进来*/
main()                      /*主函数名，函数从这里开始*/
{
    int a,b,sum;             /*定义三个整型变量，名称分别为 a、b 和 sum*/
    a=5;                    /*给 a 赋值为 5*/
    b=10;                   /*给 b 赋值为 10*/
    sum=a+b;                /*把 a+b 的值赋给 sum*/
    printf("sum is %d.\n", sum); /*输出 sum 的值*/
}
```

把上面程序输入 Turbo C/C++3.0 的编辑器，然后进行编译和运行，再转到用户输出窗口即可以看到运行结果如下：

sum is 15.

下面逐行对这个程序进行详细解说。

- 第 1 行为注释，C 语言的注释以 “/*” 和 “*/” 作为注释内容的分界符。注释可增强程序的可读性，帮助理解程序的含义。
- 第 2 行是让计算机系统把文件 `stdio.h` “包含” 进来。C 语言里有关输入和输出的函数都定义在这个文件里，要把结果输出到屏幕上，就要调用输出函数，也就需要把 `stdio.h` 文件包含进来。
- 第 3 行为程序执行的起点，即 `main()` 函数，称为“主函数”。每一个 C 程序都由一个或多个函数组成，其中必须有一个名为 `main` 的函数。
- 第 4 行的左花括号 “{” 为 `main()` 函数体的开始分界符。

- 第5行声明 a、b 和 sum 为整型变量。C 语言在使用变量之前必须先声明其类型。
- 第6行是一个赋值语句，把整数 5 赋给 a。
- 第7行也是一个赋值语句，把整数 10 赋给 b。
- 第8行把运算表达式 a+b 的计算结果赋给 sum。
- 第9行是调用系统函数 printf()，按照双引号内的指定格式输出结果，在 %d 这个位置上用 sum 的值代替，引号内的其余字符照原样输出，\n 是换行符，告诉计算机在输出完结果后将光标移到下一行的开头。
- 第10行右花括号 “}” 是 main() 函数的结束分界符。

看过上面这个简单的 C 程序例子后，现在来了解一个 C 程序的基本架构。

1. #include 宏命令及头文件

在程序中加入 #include <file name> 时，C 编译系统会自动把 <> 内的头文件的内容全部包含进来，并且放置在 #include 这行的位置上。像上面例子中，使用了 “#include <stdio.h>”，就是将 stdio.h 的内容全部放进到该程序的开头位置。

有时我们需要将已经编写好的另一个程序包含到本程序中，或者在程序中需要调用系统库函数，这时就需要使用 #include 命令。上例中由于使用系统标准输出函数 printf()，所以要将含有该函数声明信息的 stdio.h 文件包含进来。

2. main() 函数

一个 C 程序由一个或多个函数构成，其中必须有且只有一个主函数 main()，一个函数可以调用其他函数，如上面例子中 main 函数调用 printf 函数。所有的 C 程序都是从 main 函数开始执行，并在 main 函数中结束。

3. 花括号 {}、语句块和函数体

C 程序以函数为单位，编写 C 程序就是编写一个个函数。每个函数的主体都包括在一对花括号 “{}” 内。有时把完成某个功能或操作的几个语句用一对花括号括起来，组成一个语句块（也叫复合语句）。若一个函数体内有多对花括号，则最外层的一对是函数体的分界符。

4. 变量

变量是一个 C 程序不可缺少的最基本的部分，是用来存放数据的。概括地说，一个计算机程序包括两个部分，第一部分是对数据的描述，第二部分是对操作的描述。变量就是用来描述数据的。C 语言的变量必须先声明后使用，声明时要说明变量的数据类型和名字。C 语言中的数据类型有整型 (int)、浮点型 (float)、字符型 (char) 等。

5. 语句

每个 C 函数由若干个语句构成，每个语句以 “;” 结束，可以一行书写多个语句。C 语言有表达式语句、函数调用语句、控制语句、变量定义语句等。如上面 main() 函数体有 5 个语句，第 1 个语句为变量定义语句，第 2~4 语句为赋值（表达式）语句，第 5 个语句为函数调用语句。

6. 输入和输出

每个程序必须有 0 个或多个输入，有至少一个输出。如果一个程序没有输出，就无从知道它的运算和操作结果。C 语言的输入和输出是通过调用系统提供的标准输入输出函数实现的。如上例中调用 printf() 函数以指定格式输出计算结果。上例中的 “printf(“sum is %d\n”, sum);” 指定以双引号内的格式输出，在 %d 处代入 sum 的值，双引号内的所有其他字符照原样输出，“\n” 是换行符。

关于输出函数 printf() 和输入函数 scanf() 的详细介绍请参看 3.4 节。

1.3.3 C 程序中的单词

就像单词是自然语言的最小单位一样，单词也是计算机程序语言的最小单位。只是程序语言的单词与自然语言的单词不同而已。C 语言的单词可分为标识符、关键字、分界符、空白、和运算符几类。

1. C 语言的标识符

在 C 语言中，变量名、函数名、符号常量或者标号称为用户标识符。C 语言的标识符由英文大小写字母、数字或下划线组成，首字母必须是英文字母或下划线。标识符的长度没有限定，决定于使用的编译系统，如 Turbo C 允许标识符长度为 32 个字符。通常取前 8 个字符为有效字符，所以给变量或函数取名时最好保证前 8 个字符能够区别不同，例如 `numberofstudent` 和 `numberofteacher` 在某些系统中被认为是同一个变量。另外，C 语言认为一个字符的大写形式与小写形式代表不同意义，例如 `Name` 和 `name` 被认为是两个不同的标识符。还有要注意的是，系统的保留字（又称关键字，见表 1-1）是不可以用作标识符的。

- 下面是合法的标识符：

`Min, t, SIZE, M1, x10, count_1, _myargument`

- 下面是不合法的标识符：

`for` 使用关键字

`8h` 以数字开头

`-5` 以减号开头

`A*` 出现非法字符*

`x<y` 出现非法字符<

- 下面是三个不同的标识符：

`count, Count, COUNT`。



注意

标识符虽然是由程序员随意定义，但标识符是用于标识某个量的符号。因此命名时应尽量有相应的含义，增加程序的可读性，做到“见名知意”。

2. C 语言的关键字

标识符是用户用来命名变量、标号和函数的文字，而关键字是编译系统本身提供给用户使用的标识符，用户在进行编程时必须按照系统规定的方式使用这些关键字，而不能更改或者重新定义它们。用户自定义的变量名或函数名等标识符不能与关键字相同，如上面的 `for` 是关键字，所以不可用作标识符。C 语言提供的关键字共有 32 个，如表 1-1 所示。

表 1-1 C 语言的关键字

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

**注意**

C语言是大小写敏感的！所有关键字都是小写的，若把某个或全部字母写成大写形式，则系统不把它当做关键字，而当成普通标识符。

3. C语言的分界符

C语言的分界符是用来分隔特定的语法单位的。例如，花括号“{}”用来界定函数体的范围；圆括号“()”用来界定函数的参数；中括号“[]”用来界定数组的下标；“/*”和“*/”用来界定注释内容的范围；分号“;”用来表示一个语句的结束，等等。

**注意**

为避免遗漏必须配对使用的符号，例如注释符号、函数体的起止标识符（花括号）、圆括号等，在输入时，可连续输入这些起止标识符，然后再在其中进行插入来完成内容的编辑。

4. C语言的空白

一个C程序除了必需的关键字和用户自定义的标识符以外，一般还需要有一些辅助性的字符，如空格(space)、制表符、回车、换行、注释等，这些字符起到使程序的版面的层次、段落分明，反映出程序的逻辑结构，从而提高程序的可读性的作用。这些字符在编译时不会被生成任何代码，所以称它们为空白(blank)。

5. C语言的运算符

运算符在表达式中表示对数据执行某类运算。C语言的运算符非常丰富，共有13类34种，其中最常用到的有，算术运算符：加(+)、减(-)、乘(*)、除(/)、求余(%) 5种，关系运算符：大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)、不等于(!=) 6种，赋值运算符(=)，等等。各种运算符的作用将分散在后面各章节分别作详细介绍。

**注意**

C语言的等于运算符由两个“=”组成，即“==”，而赋值运算符是一个“=”。千万别写错！

1.3.4 如何编写可读性良好的C程序

1. 命名风格

很多初学者习惯采用单个字母或没有任何意义的随意的字母组合来命名变量或函数。但是为了提高程序的可读性（让人容易读懂它），建议读者养成良好的标识符的命名习惯，尽量遵循如下原则：

- 含义明确。选用能表达明确意义的英文单词命名。如count、name、number等。
- 规范缩写。采用规范的单词缩写形式：首字母一定要保留、首部比尾部重要、辅音字母比元音字母重要等。如stu、adr、id、cmp等。
- 规范分隔。采用规范的单词分隔形式。如is_equal、id_card、stu_name等。
- 限制长度。各类标识符长度尽量控制在8个字符之内（不同的C编译系统允许的标识符长度不同，Turbo C为31个字符）。
- 大小写约定。符号常量名采用全部大写的标识符；变量名和函数名采用全部小写的标识符。

2. 注释风格

C语言的注释格式为：/* */，注释时要注意以下几点：

- “/” 和 “*/” 必须成对使用，且 “/” 和 “*”、以及 “*” 和 “/” 之间不能有空格，否则都出错。
- 注释的位置，可以单占 1 行，也可以跟在语句的后面。
- 如果一行写不下，可另起一行继续写。
- 注释中允许使用汉字。在非中文操作系统下，看到的是一串乱码，但不影响程序运行。
- 编写程序时先编写注释、后编写代码；修改程序时先修改注释、后修改代码。
- 注释不仅仅是只列出代码执行的功能，而且还应给出为什么如此编写代码的原因或理由。

3. 布局风格

一个程序的可读性与它的版面布局有很大关系。程序的版面布局与采用哪一种花括号的格式影响，也离不开前面提过的空白的使用。应适当地采用缩进和分隔符，使程序逻辑结构清晰、段落层次分明。下面介绍常见的几种花括号格式以及空白的使用：

(1) 花括号的格式。

- 1TBS

这种格式源自 C 语言的始祖 K&R 的名著《The C Programming Language》。如下面代码段所示：

```
for(i=1;i<=100;i++){  
    sum=sum+i;  
}
```

- Allman

这种格式源自 Eric Allman 编写 BSD 软件采用的格式。如下面代码段所示：

```
for(i=1;i<=100;i++)  
{  
    sum=sum+i;  
}
```

- Whitesmith

这种格式源自 Whitesmith C 编译器。如下面代码段所示：

```
for(i=1;i<=100;i++)  
{  
    sum=sum+i;  
}
```

- GNU

这种格式如下面代码段所示：

```
for(i=1;i<=100;i++)  
{  
    sum=sum+i;  
}
```

上面几种格式各有优缺点：第 1 种格式优点是节省纵向空间，缺点是不容易找到 “}”；第 2 种格式优点是块结构清晰，层次分明，缺点是占用纵向空间较多；第 3 种格式和第 2 种差不多，只是花括号与其包括的代码联系更紧密；第 4 种格式可读性最高，可是占用的空间也最多。

可以根据自己的喜好选用上面的任何一种格式，不过在同一个软件中最好采用同一种格式。

(2) 空白的使用。

前面已经提过，在一个程序中适当地使用空白可提高程序的可读性。下面重点说明一下