

高等院校计算机教育系列教材

C语言程序设计教程

(第2版)

许 勇 主 编
戴仕明 主 审



清华大学出版社

高等院校计算机教育系列教材

C 语言程序设计教程(第 2 版)

许 勇 主 编
戴仕明 主 审

清华大学出版社

北 京

内 容 简 介

本书讲授 C 语言程序设计的基本思想、方法和解决实际问题的技巧。在体系结构安排上,尽可能将概念、知识点与案例相结合,进行案例教学,并且每一章都精心设计了习题与上机实验(附习题和上机实验答案),突出实用性、操作性。

全书共分 13 章,主要内容包括:C 语言概述、C 语言程序设计的初步知识、基本算法、顺序结构、选择结构、循环结构、数组、函数、指针、编译预处理、复杂数据类型、位运算、文件等。

本书可作为高等学校各专业 C 语言程序设计课程的教材,也可以作为等级考试的辅导教材,以及计算机爱好者自学用书和各类工程技术人员的参考书。

本书配有电子教案,并提供程序源代码,以方便读者自学。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C 语言程序设计教程(第 2 版)/许勇主编;戴仕明主审. —北京:清华大学出版社,2006.11
(高等院校计算机教育系列教材)

ISBN 7-302-14017-0

I. C… II. ①许… ②戴… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 121556 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

组稿编辑:章忆文

文稿编辑:杨作梅

排版人员:阚道燕 房书萍

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:21.25 字数:503 千字

版 次:2006 年 11 月第 2 版 2006 年 11 月第 1 次印刷

书 号:ISBN 7-302-14017-0/TP·8421

印 数:1~4000

定 价:29.00 元

丛 书 序

人类进入新千年时, 计算机科学已成为一块极其活跃的、崇尚发展与创新的领地并成为我们这一时代决定性的先驱技术。计算机是现代文化构成不可或缺的组成部分, 是现代科学技术发展的先导, 并成为世界经济巨量增长的根本引擎。同时, 计算机技术的发展日新月异, 它的快速发展对计算机科学与工程的教育产生了深刻的影响。因此, 计算机教育在世界各国倍受重视, 计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

伴随着计算机新技术的不断涌现, 已有技术几年内即变得陈旧。与此同时, 计算机教育也被其所在的专业、文化和社会范围的改变影响着。计算机学科已变得更宽广、内容更丰富, 其应用领域不断飞速增长。因此, 无论在教学体系、教学内容还是教学方法、教学手段上都必须进行深化改革, 与时俱进。CC2001 的出现, 无疑是对计算机学科课程体系一个崭新的完整的引导。我们工科院校无论计算机专业还是非计算机专业的计算机科学与工程的教育应该紧密有机地与工科学生的培养目标相结合, 注重知识、能力、素质教育三方面的综合教育, 加强计算机系统的认知、分析、设计和应用能力、算法设计与分析能力和程序设计能力以及计算思维能力等方面的培养。

原化工部部属高校计算机教育协作组结合工程教育的特点, 大力开展计算机教育协作与交流, 十年来开展了务实的、全方位的、卓有成效的教学研讨及教研观摩等活动, 极大地促进了交流并推动了各校计算机教育的发展。同时, 协作组不断地扩展, 吸收了许多其他领域的高校参加, 共同为我国计算机高等教育事业的发展与完善进行广泛的交流探索。

目前参加这个协作组的主要高校有:

清华大学	南京大学	天津工业大学	北京化工大学
南京工业大学	青岛科技大学	郑州大学	武汉化工学院
沈阳化工学院	南京师范大学	华南理工大学	河北行政学院
南京工程学院	淮海工学院	北京石油化工学院	江苏石油化工学院

在清华大学出版社的大力支持下, 本协作组 2001 年年会决定组织出版一套最新的计算机系列教材, 第一期出版 10 部有关程序设计与软件应用方面的教材。它们是: 《计算机导论》、《C 语言程序设计》、《Visual Basic 语言程序设计》、《Java 程序设计》、《面向对象程序设计——C++》、《SQL Server 数据库原理及应用教程》、《C#编程及应用程序开发教程》、《组网技术与配置》、《现代语音技术基础与应用》和《计算机图形学基础教程》等。

本系列教材依据 CC2001 框架, 精心策划、准确定位, 概念清晰, 例题丰富, 深入浅出, 内容翔实, 体系合理, 重点突出, 是一套面向高等学校计算机和非计算机专业学生的

计算机基础与应用系列教材, 也可供从事计算机应用和开发的各类人员学习使用。

本系列教材源于十几所全国重点大学和普通高等院校计算机教育的教学改革与实践, 凝聚了工作在教学第一线的任课教师的教学经验与研究成果。我们期望本系列教材的出版, 并在教学实践中的不断完善与更新, 为我国高校计算机教育事业做出新的贡献。

编委会

编委会名单

主编: 朱群雄

编委: 闵华清	王晓峰	邵定宏
刘川来	彭四伟	刘 斌
刘新民	张彦锋	吕纪国
刘 烨	王相林	蔡莲红
孙正兴	许 勇	戴仕明

前 言

C 语言是一种在世界范围内被普遍采用的优秀的程序设计语言，是现代最流行的通用程序设计语言之一，功能丰富，使用灵活，可移植性好，深受广大用户欢迎。C 语言的数据类型丰富，既具有高级程序设计语言的优点，又具有低级程序设计语言的特点；既可以用来编写系统程序，又可以用来编写应用程序。因此，C 语言正在被迅速地推广和普及。

本教材通过大量实例，系统地介绍了 C 语言的语法结构。全书共分 13 章和 4 个附录，第 1 章为 C 语言概述，主要内容包括 C 语言的发展、特点和 C 程序的组成和执行以及上机实验。第 2 章为 C 语言程序设计的初步知识，主要内容包括数据类型、数据类型转换、运算符与表达式以及上机实验。第 3 章为基本算法，主要内容包括程序与算法、算法的特征、算法的表示和程序设计基本算法。第 4 章为顺序结构，主要内容包括 C 语句、数据的输入与输出、顺序程序设计以及上机实验。第 5 章为选择结构，主要内容包括关系运算符和关系表达式、逻辑运算符与逻辑表达式、选择结构语句、选择结构程序设计举例和上机实验。第 6 章为循环结构，主要内容包括 while 型循环、do-while 型循环、for 型循环、循环的嵌套、其他控制语句、循环结构综合实例和上机实验。第 7 章为数组，主要内容包括一维数组、二维数组、多维数组介绍、字符数组、数组综合实例和上机实验。第 8 章为函数，主要内容包括函数概念引入、库函数、函数定义、函数间的参数传递、函数调用、数组与函数变量的作用域、变量的存储类型、函数的存储分类以及上机实验。第 9 章为指针，主要内容包括指针的含义、指针变量、指针与数组、指针与函数、指针与字符串、字符数组与字符指针的区别、指向指针的指针、指针类型小结、综合实例和上机实验。第 10 章为编译预处理，主要内容包括文件包含、宏定义、条件编译和上机实验。第 11 章为复杂数据类型，主要内容包括结构体、用结构指针操作链表、共用体、枚举类型、类型定义 typedef 和上机实验。第 12 章为位运算，主要内容包括二进制位运算概述、位运算符、位段和上机实验。第 13 章为文件，主要内容包括 C 文件的概述、文件的打开与关闭、文件的读写、文件的定位与出错检测和上机实验。附录 A 为运算符的优先级及其结合性，附录 B 为常用 ASCII 代码对照表，附录 C 为各章习题参考答案，附录 D 为各章上机实验参考答案。

本教材组织精练，例题简单，容易理解，并配备了各种类型的练习和实践操作题，便于学习掌握和上机练习。对于 C 语言中重要和较难理解、容易出错的内容，书中均特别加以强调和说明。此外，在介绍 C 语言的语法结构的同时，也强调了计算机算法和结构化设计方法的概念和作用。本书后附有习题和实验题的参考答案，可供读者学习时参考。本书可作为高等学校 C 语言程序设计课程的教材，也可作为计算机等级考试的辅导教材。

本书以 ANSI 标准 C 语言为背景，有关内容不依赖于任何具体的 C 系统。本书虽然介绍了国内使用较多的 Turbo C 系统，但读者仍可自由选用其他符合 ANSI 标准的 C 系统编程环境作为学习工具。

本书由许勇担任主编，戴仕明担任主审。全书由何光明策划并确定框架结构，最后由

许勇统编定稿。唐才琴、孙玉香、杨明、杨萍、赵传审、许娟、尹晓睿、史国川、史春联、周松、赵明、陈智等同志在内容编写、程序测试、文字校对等过程中也付出了辛勤劳动,另外本书在编写过程中,得到兄弟高校从事计算机教育的老师的关心和帮助,教研室的同仁也提出了许多宝贵意见,在此一并表示衷心的感谢。

本书配有电子教案,并提供程序源代码,以方便读者自学,请到 www.wenyuan.com.cn 下载。

限于作者水平,书中难免存在不当之处,恳请广大读者批评指正。任何批评和建议请发至: Book21Press@126.com。

编 者

目 录

第1章 C语言概述..... 1	3.2 算法的特征..... 33
1.1 C语言的发展史..... 1	3.3 算法的表示..... 34
1.2 C语言的特点..... 1	3.4 程序设计基本算法..... 38
1.3 C程序的组成..... 3	3.4.1 穷举法..... 38
1.3.1 简单的C程序分析..... 3	3.4.2 迭代算法..... 38
1.3.2 简单的C程序应用..... 5	3.4.3 递推算法..... 39
1.4 C程序的执行..... 6	3.4.4 递归算法..... 40
1.4.1 C语言的编译..... 6	小结..... 41
1.4.2 Turbo C 2.0 常用菜单命令 介绍..... 7	习题..... 42
小结..... 8	上机实验..... 42
习题..... 9	第4章 顺序结构..... 43
上机实验..... 9	4.1 C语句介绍..... 43
第2章 C语言程序设计的初步知识..... 10	4.2 数据的输入与输出..... 45
2.1 数据类型..... 10	4.2.1 格式输出函数 ——printf函数..... 45
2.1.1 常量与变量..... 10	4.2.2 格式输入函数 ——scanf函数..... 50
2.1.2 整型数据..... 13	4.2.3 字符输出函数 ——putchar函数..... 53
2.1.3 实型数据..... 15	4.2.4 字符输入函数 ——getchar函数..... 54
2.1.4 字符型数据..... 16	4.3 顺序程序设计..... 55
2.2 数据类型转换..... 18	小结..... 57
2.2.1 自动转换与强制转换..... 18	习题..... 57
2.2.2 不同数据类型间的混合运算 实例..... 20	上机实验..... 61
2.3 运算符与表达式..... 21	第5章 选择结构..... 62
2.3.1 算术运算符与算术表达式..... 21	5.1 关系运算符与关系表达式..... 62
2.3.2 赋值运算符与赋值表达式..... 24	5.1.1 关系运算符..... 62
2.3.3 逗号运算符与逗号表达式..... 27	5.1.2 关系表达式..... 63
小结..... 29	5.2 逻辑运算符与逻辑表达式..... 64
习题..... 29	5.2.1 逻辑运算符..... 64
上机实验..... 32	5.2.2 逻辑表达式..... 65
第3章 基本算法..... 33	
3.1 程序与算法..... 33	

5.3 选择结构语句	67	7.4 字符数组	111
5.3.1 if 语句	67	7.4.1 字符数组定义	111
5.3.2 条件运算符	73	7.4.2 字符数组初始化	113
5.3.3 switch 语句	75	7.4.3 字符串与字符数组	113
5.3.4 标号语句和 goto 语句	77	7.4.4 字符数组的输入与输出	114
5.4 选择结构程序设计举例	78	7.4.5 字符数组处理函数	115
小结	80	7.5 数组综合实例	119
习题	80	小结	121
上机实验	83	习题	122
第 6 章 循环结构	84	上机实验	124
6.1 while 型循环	84	第 8 章 函数	126
6.1.1 while 结构的形式	84	8.1 函数概念引入	126
6.1.2 while 循环的语义	85	8.2 库函数	127
6.2 do-while 型循环	87	8.3 函数定义	129
6.2.1 结构形式	87	8.3.1 函数定义的形式	129
6.2.2 do-while 循环的含义	88	8.3.2 函数的简单应用	131
6.3 for 型循环	89	8.4 函数间的参数传递	132
6.3.1 for 循环表现形式	89	8.4.1 形式参数和实际参数	132
6.3.2 for 与 while	91	8.4.2 函数返回值	134
6.3.3 for 语句的实例	92	8.5 函数调用	136
6.4 循环的嵌套	93	8.5.1 调用的一般形式	136
6.5 其他控制语句	94	8.5.2 函数调用方式	136
6.5.1 break 语句	94	8.5.3 函数调用声明	138
6.5.2 continue 语句	95	8.5.4 函数的嵌套调用	139
6.6 循环结构综合实例	96	8.5.5 函数的递归调用	141
小结	99	8.6 数组与函数	143
习题	99	8.7 变量的作用域	145
上机实验	101	8.7.1 局部变量	145
第 7 章 数组	103	8.7.2 全局变量	146
7.1 一维数组	103	8.8 变量的存储类型	149
7.1.1 一维数组的定义与引用	103	8.8.1 变量的动态和静态存储方式	149
7.1.2 一维数组的初始化与应用	104	8.8.2 局部变量的存储类别	150
7.2 二维数组	107	8.8.3 全局变量的存储类别	153
7.2.1 二维数组定义与引用	107	8.8.4 存储类别小结	155
7.2.2 二维数组初始化	108	8.9 函数的存储分类	156
7.2.3 二维数组应用举例	109	8.9.1 内部函数	156
7.3 多维数组介绍	111	8.9.2 外部函数	157

小结	158	10.2 宏定义	218
习题	159	10.2.1 无参数宏定义	218
上机实验	164	10.2.2 有参数的宏定义	221
第9章 指针	165	10.2.3 宏替换与函数调用的区别	223
9.1 指针的含义	165	10.3 条件编译	224
9.2 指针变量	166	小结	227
9.2.1 指针变量的概念	166	习题	227
9.2.2 指针变量的定义	166	上机实验	229
9.2.3 指针变量赋值与初始化	167	第11章 复杂数据类型	230
9.2.4 指针的算术运算	169	11.1 结构体	230
9.2.5 指针变量的引用	170	11.1.1 结构体类型的声明	230
9.3 指针与数组	171	11.1.2 结构体变量	231
9.3.1 一维数组的指针	172	11.1.3 结构体变量的引用与 初始化	233
9.3.2 二维数组的指针	175	11.1.4 结构体数组	235
9.4 指针与函数	180	11.1.5 结构体指针	238
9.4.1 变量的指针作为函数参数	180	11.1.6 结构体与函数	241
9.4.2 指针型函数	183	11.1.7 结构体嵌套	244
9.4.3 一维数组的指针作为 函数参数	188	11.2 用结构指针操作链表	245
9.4.4 二维数组的指针作为 函数参数	191	11.2.1 链表概述	245
9.5 指针与字符串	195	11.2.2 动态内存管理函数	246
9.5.1 单字符串	196	11.2.3 链表的基本操作	247
9.5.2 多字符串	200	11.2.4 结点的插入与删除	251
9.5.3 字符串指针作函数参数	201	11.3 共用体	253
9.6 字符数组与字符指针的区别	202	11.3.1 共用体类型定义	253
9.7 多级指针	202	11.3.2 共用体变量的引用	255
9.7.1 定义形式	202	11.4 枚举类型	257
9.7.2 传给 main 函数的参数	205	11.5 类型定义 typedef	260
9.7.3 void 指针类型	206	小结	261
9.8 指针类型小结	207	习题	262
9.9 综合实例	207	上机实验	265
小结	210	第12章 位运算	267
习题	210	12.1 二进制位运算概述	267
上机实验	214	12.2 位运算符	268
第10章 编译预处理	215	12.3 位段	273
10.1 文件包含	215	小结	274
		习题	274

上机实验	276	13.3.4 格式化读写函数	288
第 13 章 文件	277	13.4 文件的定位与出错检测	290
13.1 C 文件的概述	277	13.4.1 文件的定位	290
13.1.1 文件的基本概念	277	13.4.2 文件操作的出错检测	292
13.1.2 文件指针	279	小结	293
13.2 文件的打开与关闭	280	习题	293
13.2.1 文件的打开	280	上机实验	296
13.2.2 文件的关闭	282	附录 A 运算符的优先级及其结合性	297
13.3 文件的读写	283	附录 B 常用 ASCII 代码对照表	298
13.3.1 单个字符的读写操作	283	附录 C 各章习题参考答案	299
13.3.2 字符串的读写操作	284	附录 D 各章上机实验参考答案	316
13.3.3 数据块的读写操作	286		

第 1 章 C 语言概述

计算机本身是无生命的机器，要使计算机能够运行起来，完成各种人们预想的任务，就得编制相应的程序，让计算机来执行。而这些程序都是由语言编写而成的。

随着计算机技术的不断发展，计算机程序设计的研究开发也得到迅猛的发展。在各种各样的高级语言中，C 语言以其强大的功能和优良的特点，成为国际上公认的最重要的少数几种通用程序设计语言之一。

1.1 C 语言的发展史

C 语言由 Dennis Ritchie 设计，并首次在一台使用 UNIX 操作系统的 DEC PDP-11 计算机上实现。C 语言是由一种早期的编程语言 BCPL 发展演变而来的，BCPL 语言目前在欧洲还在使用。Martin Richards 改进了 BCPL 语言，从而促进了 Ken Thompson 所设计的 B 语言的发展，最终促进了 20 世纪 70 年代 C 语言的问世。

后来，又对 C 语言做了多次的改进，但主要还是在贝尔实验室使用。直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们的普遍关注。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”，使 C 移植到其他机器时所需做的工作大大简化，这也推动了 UNIX 操作系统迅速地在各种机器上的实现。例如：VAX、AT&T 等计算机系统都迅速相继开发了 UNIX。随着 UNIX 使用的日益广泛，C 语言也迅速得到推广。C 语言与 UNIX 可以说是孪生兄弟，在发展过程中相辅相成。1978 年以后，C 语言已先后移植到大型、中型、小型及微型机上，并独立于 UNIX。现在 C 语言已风靡全世界，成为世界上应用最广的几种计算机语言之一。

随着微型机的日益普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准协会为 C 语言制定了一套 ANSI 标准，成为先行的 C 语言标准。

除了系统软件外，C 语言还成功地用于数值计算、文字处理、数据库、计算机网络和多媒体等。C 语言所呈现的高级语言强有力的表达能力和效率，使得它成为近 10 年来在计算机程序设计实践中做出重大贡献的一种语言。

目前，在计算机上广泛使用的 C 语言编译系统有 Microsoft C，Turbo C，Borland C 等，虽然它们的基本部分是相同的，但还是有差异，本书的上机环境采用 Turbo C。

1.2 C 语言的特点

作为目前应用最广泛的高级程序设计语言，C 语言有如下明显的特点。

1. C 语言的优点

(1) 语言描述简洁、灵活、高效

C 语言只有 32 个标准的关键字和 9 种控制语句, 并且以易读易写的小写字母为基础, 压缩了一切不必要的成分, 使程序书写规整紧凑。

(2) 有丰富的数据类型

C 语言数据类型丰富, 不仅具有 4 种最基本的数据类型(char, int, float, double), 还有多种组合类型(数组和结构)以及复杂的导出类型, 并允许使用简单的组合结构构造复杂的数据类型或干脆由用户自己定义数据类型。

(3) 运算符丰富

C 语言提供了 45 种标准的运算符和多种获取表达式值的方法, 并提供了与地址密切相关的指针及其运算符。C 语言的运算符不仅具有优先级的概念, 还有结合性的概念。因此, 灵活使用各种运算符和表达式, 不仅可以简化程序, 还可以实现在其他语言中难以实现的运算。

(4) C 语言具有固定的标识符

C 语言的标识符主要用来表示常量、变量、函数和类型等的名字, 它们只是起标识作用的一种符号。C 语言共有 32 个固定的标识符, 它们都用小写字母表示, 如图 1.1 所示。

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

图 1.1 C 语言的标识符

(5) 提供了功能齐全的函数库

C 语言标准函数库提供了功能极强的各类函数库, 例如: 串、数组、结构乃至图形的处理等, 只需要调用一下库函数即可实现, 为编程者提供了极大的方便。

(6) 具有结构化的控制语句

C 语言提供了 9 种控制语句实现 3 种结构(顺序、分支和循环结构)。如 if-else, while, switch, for 等。并用函数作为程序模块, 是理想的结构化程序设计语言。C 程序由具有独立功能的函数构成, 函数定义是平行、独立的, 但函数调用可以是嵌套调用和递归调用, 通过函数调用可以实现复杂的程序功能, 多种存储属性的数据可共享同一段内存, 从而保证了模块化的程序设计风格。另外, 对于复杂的源程序文件, 可以分割成多个较小的源文件, 分别编译和调试, 最后组装, 连接得到可执行的目标程序文件。

(7) 具有良好的通用性和程序的移植性

在 C 语言中, 与硬件有关的操作都是通过调用系统提供的库函数实现的, 这使得 C 语言具有很好的通用性, 程序能容易地从一种计算机环境移植到另一种计算机环境中。

(8) 生成目标代码质量高, 程序执行效率高

C 语言生成目标代码的效率与汇编语言相比, 一般低 10%~20%。C 语言也可以像汇编语言一样对位、字节和地址, 甚至对硬件进行直接操作。换句话说, C 语言既具有汇编语言的强大功能, 又没有汇编语言的难度, 特别适合做底层开发。

(9) 语法限制不严格

C 语言程序设计的自由度大, 例如对数组下标越界不作检查, 各种类型的变量可以通用等。

2. C 语言的缺点

当然, C 语言的缺陷还是十分明显的。C 语言缺乏一致公认的标准, 表现在语法限制不太严谨, 运算符的优先级和结合性比较复杂, 不容易记忆。C 语言对数据类型缺乏一致性的检测, 对数组、结构等类型的整体运算存在一定的限制。C 语言在程序方面灵活性有余, 而安全性和可靠性不足。

对于上述问题, 计算机科学家和工程技术人员正在不断寻求解决的办法, 并不断提出改进的方案。

1.3 C 程序的组成

1.3.1 简单的 C 程序分析

下面介绍一个最基本 C 程序来说明 C 程序的特性。

例 1.1 显示: “What is your name?”。

```
main()                                /*主函数*/
{
    printf("What is your name? \n");    /*调用库函数 printf() 显示字符串*/
}
```

运行结果:

What is your name?



说明: 本程序由一个 main() 函数构成。main 是函数名, 函数名后面的括号内是填写参数的, 由于本程序主函数没有参数, 所以是空的, 但括号不能省略。main() 后面有一对花括号, 花括号内由语句组成, 语句末尾一定加上分号。本程序只有一个语句。

printf() 函数是 C 语言的库函数, 它的功能是在屏幕上输出双引号所包含的内容, “\n” 是转义字符, 它代表回车换行。

程序的注释部分应在 “/*” 与 “*/” 之间, “/” 和 “*” 之间不能有空格, 注释部分允许出现在程序的任何位置。

下面我们再分析一个通用的 C 程序。

例 1.2 用户输入两个整数, 计算机输出两个数的和。

```
main()                                /*主函数*/
{
    int x, y, z;                        /*定义变量*/
```

```

scanf("%d,%d",&x,&y);          /*从键盘上输入两个变量*/
z=add(x,y);                    /*计算两个数的和值*/
printf("The result is %d",z);  /*输出*/
}
int add(int x,int y)
{
    int z;
    z=x+y;                    /*计算和值*/
    return(z);
}

```

输入: 3,5

运行结果:

The result is 8



说明: 这个程序包括两个函数,一个是主函数,另一个 add 函数是求出两个整数的和。在执行时,先由 scanf() 函数从键盘上输入两个数字,将输入的数传入 add() 函数中,在 add() 函数中计算两个数的和值,并用 return 语句将和值作为函数的返回值。

以上实例及说明体现了一般 C 语言程序的如下特点:

- 程序至少要有有一个 main 函数,用户也可以自己根据需要设计函数,像上面的 add() 函数。
- 函数由两部分组成。第一部分是函数的说明部分,如函数的名称,函数的返回值类型,函数的参数及类型。第二部分是函数的实现部分,包括变量定义和执行语句。
- C 程序总是从 main 函数开始执行,不论 main 函数在程序的什么地方,也就是说,可以将 main 函数放在任何位置。
- C 程序的书写比较自由,可以在一行上写若干语句,也可以在多行上写一条语句。但每个语句后面都要有一个分号,这个分号是必不可少的。这一点必须注意。
- C 语言中没有专门的输入、输出语句。输入和输出是通过 scanf 和 printf 两个库函数实现的。
- C 程序中可以用“/*.....*/”对任何部分进行注释,好的程序都要有必要的注释以提高程序的可读性。

从例 1.2 可以推出一般 C 语言程序的构成如下:

```

函数类型 函数名(参数类型 参数名) /*自定义函数*/
{
    函数体;
}
main()
{
    变量定义部分;
    语句执行部分;                /*语句执行部分包括调用自己定义的函数*/
}

```



说明： C语言的源程序由函数构成。每一个函数完成相对独立的功能，其中至少包括一个主函数(main()函数)。

一个自定义函数由两部分组成：

① 函数的首部，即函数的第一行。包括函数名、函数类型、参数类型和参数名。

例如：

```
int add(int x,int y);
```

int
add
(int
x,
int
y);

↓
↓
↓
↓
↓
↓

函数类型
函数名
函数参数类型
函数参数名
函数参数类型
函数参数名

一个函数名后面必须跟一对圆括号，函数参数可以没有。

② 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{}为函数的范围。

函数体一般包括：

声明部分：在这部分中定义所用到的变量，如 add()函数中的 int z。

执行部分：由若干个语句组成，如 add()函数中的 z=x+y。当然，在某些情况下也可以没有声明部分。甚至可以既无声明部分，也无执行部分。例如：

```
add()
{ }
```

它是一个空函数，什么也不做，但这是合法的。

1.3.2 简单的C程序应用

首先，书写程序应从书写清楚，便于阅读、理解和维护的角度出发，在书写程序时应遵循以下规则：

- 说明一条语句占一行。
- 在{}中的部分，通常表示程序的某一层结构。{}一般与该结构语句的第一个字母对齐，并单独占一行。
- 低一层次的语句说明可比高一层的语句说明缩进若干格后书写。以便看起来更加清晰，增加程序的可读性。

在编程时应力求遵循这些规则，以养成良好的编程风格。

例 1.3 由键盘输入两个整数，计算出它们的和与差，并在屏幕上输出。

```
main()
{
    int a,b,c,d;
    scanf("%d%d",&a,&b);
    c=a+b;
    d=a-b;
```

```
printf("\n%d+%d=%d\n%d-%d=%d\n",a,b,c,,a,b,d);
}
```

本例也只有一个 main 函数。首先定义变量为整型,接着调用 scanf 函数,从键盘上输入两个整数,存入变量 a 和 b 的存储单元中,然后求和与差数,并分别存入变量 c 和 d 的存储单元中,最后执行 printf 语句输出计算结果。

输入: 8 3

运行结果:

8+3=11

8-3=5

1.4 C 程序的执行

1.4.1 C 语言的编译

C 语言的编译是把源程序转换为二进制的可执行代码。一般需要以下几个步骤。

(1) 首先进入 C 语言集成开发环境 Turbo C 2.0, 如图 1.2 所示。

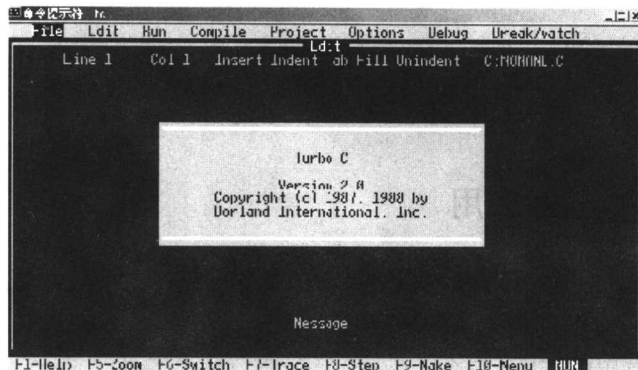


图 1.2 Turbo C 2.0 的启动界面

(2) 在开发环境中编辑源文件,编好后可以存盘(文件名为:文件名.c)。注意这时的文件是 ASCII 文件,不能执行。

(3) 编译。编译就是将已经编好的源程序翻译成二进制的目标代码,如果有错误,会给出错误提示,这时应该重新进入编辑环境,对源程序进行修改,直到通过编译为止。编译后的文件的扩展名为.obj。经过编译的二进制代码不能直接执行,因为一个程序的各个模块往往是单独编译的,必须把经过编译的各个模块的目标代码和系统提供的标准模块进行连接后才能够执行。

(4) 连接。经过连接后,源程序成为可执行的文件,它是计算机能够直接运行的文件。文件的扩展名为.exe。

(5) 执行。直接在操作系统的环境下输入文件名就可以执行经过编译连接的可执行文