

天津市高等学校计算机基础课程规划教材

C程序设计教程

(第2版) 高福成 编著 边莫英 主审

天津市高等学校计算机基础课程规划教材

C 程序设计教程

(第 2 版)

高福成 编著

边奠英 主审



内容提要

本书以程序设计为主线,以程序设计的需要带动语言知识的学习,系统介绍了C语言及其程序设计技巧。全书共十章,包括C程序设计的初步知识、基本数据类型、数据运算、程序流程控制、数组和字符串、指针、C函数、复合数据类型、文件、编译预处理和分段编译,并通过丰富的程序设计实例,详细介绍了算法知识。各章都编排了大量的练习题,以帮助读者在初步掌握C语言的基础上,着重培养程序设计能力。

本书可作为高等学校本科生C语言程序设计课程教材,也可作为全国计算机等级考试培训教材及考前复习参考书。

图书在版编目(CIP)数据

C程序设计教程/高福成编著.—2版.—天津:天津大学出版社,2004.7

ISBN 7-5618-1927-7

I . C… II . 高… III . C语言-程序设计-教材 IV . TP312

中国版本图书馆CIP数据核字(2004)第027350号

出版发行 天津大学出版社
出版人 杨风和
地 址 天津市卫津路92号天津大学内(邮编:300072)
网 址 www.tjup.com
电 话 发行部:022-27403647 邮购部:022-27402742
印 刷 天津市宝坻县第二印刷厂
经 销 全国各地新华书店
开 本 185mm×260mm
印 张 19.25
字 数 481千
版 次 2004年7月第1版
印 次 2004年7月第1次
印 数 1—4 000
定 价 27.00元

序 言

中国要振兴,归根到底要靠我们中国人自己努力奋斗,要靠我们的全体劳动者创造出数十倍于今天的劳动生产率。这是一个全体国民素质提高的过程,人们必然要寄希望于教育。教育,特别是学校教育,是为几年乃至几十年之后的社会需求培养人才,所以教育必须面向未来。

要搞好教育,有许多事情要做,其中一条就是教材建设。面对已经到来的信息社会,学校课程到底应该让学生具备哪些基本素质,实现哪些发展,这是当前课程改革的一个重要问题。20世纪末国家提出的课程计划明确了“促进学生个性健康发展”的目标,重视认知与情感的统一、知识与能力的统一、主体精神与社会责任的统一,强调学生的素质发展,强调学生的探索创新能力、实践能力、学习能力和信息素养。

为适应课程目标的变化,需要重新审视课程内容,要删除陈旧过时的内容,吸收科学和文明发展的新成果,反映本学科最新发展动态。

要编写出课程内容具有科学性、系统性和先进性,符合本课程内在逻辑体系和学生认知规律,表达形式符合国家有关规范标准的教材,不是一件易事。为此,我们邀请了本市各高校长期从事计算机基础课教学的教师组成写作班子。这些老师们认真总结了“九五”规划教材的编写经验,反复讨论新制定的教学大纲,把课程内容有机地组合起来,把基本概念、基本原理和基本技能提炼出来,形成一个具有逻辑性、系统性的知识系统,使之有利于学生对知识的理解与迁移。

这套教材的出版,旨在推动我市高校计算机基础教育活动,提高大学生计算机基础知识水平和应用能力。我们殷切地希望广大学生、教师和专家提出宝贵意见,以便再版时修改补充。

这套教材在编写出版过程中,得到了各方人士的大力支持和帮助,特别是天津大学出版社始终给予积极配合。在此,我们一并表示衷心的感谢。

天津市普通高等学校计算机基础课程教学指导委员会
2004年4月

前 言

C语言是计算机程序设计语言中的小字辈,但卓越的性能使它风靡全世界,成为最受欢迎的语言之一。当今流行的面向对象语言 C++ 及 Internet 网上的 Java 语言就源于 C,要学习 C++ 或 Java,先要掌握 C。

目前许多高等院校都已开设 C 语言课程。参加计算机等级考试的应试者中,选择 C 语言的人数不断增加。从这几年全国计算机等级考试的情况看,许多应试者缺少 C 语言的正规训练,C 语言的精华和要害部分常常掌握不好。为了适应形势的需要,为了尽快培养一批熟练掌握 C 语言及其程序设计技术的人才,天津市教委组织编写了本书。

在计算机日益普及、计算机技术日新月异、新型软件大量涌现的今天,程序设计仍然是每一个当代大学生的基本功,是计算机素质教育的重要一环。尤其对那些希望用计算机解决本专业领域实际问题的有识之士,更需要加强程序设计训练。

学习计算机语言的惟一目的是应用,而应用要通过程序设计来体现。程序设计发展到今天,已经由技艺推进到科学,有了自己的一套基本原理和方法。进行程序设计,需要很强的逻辑思维能力,是一种极富创造性的智力劳动。对初学者而言,最使他们望而生畏的,往往也正是这一点。可以这样认为,语言是一种技能,程序设计是一门科学。因此,任何计算机语言及其程序设计的基本特点就是理论性和实践性并重,教学上应强调科学训练与技能培养并存。基于这一认识,本书以应用为目的,以提高程序设计能力为目标,以程序设计方法学为依据,系统介绍了 C 语言及其程序设计技术,把程序设计作为科学来讲授,把语言作为技能来培养,让读者在大量的程序设计实践中自然而然地熟悉和掌握语言。

本书从选材上注意了以下几点。

(1)立足基点掌握制高点。基点是会用语言来进行程序设计,学会一大批基本的程序设计单元;制高点是掌握程序设计的基本思维方法和一定的技巧。因此,本书对语言本身的问题点到为止,不深加追究,但对程序设计则根据实用和适用的原则,精选大量的例题加以阐述。这些例题不是围绕语言展开,而是服从程序设计需要的实际题目。

(2)突出重点分解难点。本书的重点章节是第 4、6、7、8 章。书中对此进行了较大篇幅的阐述。对难点问题,如指针、数据在函数间的传递和库函数的使用等,则把它们分散在多个章节内讨论。

(3)为了使读者能及时考查自己的学习效果,每一章都安排了大量的习题,包括选择题、填空题、编程题三种题型,可满足读者参加等级考试、发挥程序设计的创造性以及上机实践的需要。

根据C语言自身的特点,本书从内容上做了如下安排:第1章是C程序设计的初步知识,尽管简单,但它是学习C语言的纲;第2、5、8章从易到难介绍C语言的各种数据结构;第3章专门介绍C语言的数据运算;第4章集中介绍C语言的控制结构和程序设计的基本思维方法,体现结构化程序设计的特点;第7、9章介绍函数和文件,这是模块化程序设计的需要;第6章介绍指针,这是C语言的精华和特色,也是学习的难点;最后一章介绍了C语言环境中的特殊而较为深入的问题,这些问题基本不影响本书对初学者的要求,但对较高层次的学员来说又是必要的资料,可以作为自学之用;附录部分给出了C语言的运算符集、ASCII代码表、Turbo C 2.0函数库和编译错误信息,便于读者查阅。

本书的对象既可以是初学者,也可以是有一定语言基础的学员。除了最后一章可作为自学内容外,其余九章对于初学者,约需讲授40学时;对于具有其他计算机语言基础的学员,约需讲授30学时。另外需要36学时的上机实践(课内18学时,业余18学时)。

边莫英教授精心审阅了全部书稿。本书在编写和出版过程中,得到天津市教委和天津商学院的大力支持,天津大学出版社的编辑为此付出了大量的辛勤劳动,在此一并表示感谢。

书中错误和不妥之处,敬请广大读者不吝指正。

编著者 谨识

目 录

第 1 章 C 程序设计的初步知识	(1)
1.1 C 语言的产生和发展	(1)
1.2 C 程序的基本结构和书写风格	(2)
1.3 简单的 C 程序设计	(5)
1.4 C 程序的开发过程	(8)
1.5 Turbo C 2.0 集成环境的使用	(9)
习题一	(13)
第 2 章 基本数据类型、常量和变量	(15)
2.1 基本数据类型	(15)
2.2 常量及其类型	(17)
2.3 变量的定义及初始化	(20)
2.4 符号常数	(23)
2.5 不同类型数据的输入输出	(25)
习题二	(33)
第 3 章 数据运算	(37)
3.1 算术运算	(37)
3.2 赋值运算	(42)
3.3 逗号运算(顺序运算)	(44)
3.4 关系运算和逻辑运算	(45)
3.5 测试数据长度运算符 sizeof	(49)
3.6 位操作	(50)
3.7 常用数学函数	(55)
习题三	(56)
第 4 章 程序流程控制	(61)
4.1 结构化程序设计的三种基本结构	(61)
4.2 选择结构程序设计	(62)
4.3 循环结构程序设计	(74)
4.4 转移控制语句	(82)
习题四	(86)
第 5 章 数组和字符串	(95)
5.1 数组的概念	(95)
5.2 数组的定义和初始化	(96)
5.3 数组的基本操作	(100)
5.4 数组的应用	(105)
习题五	(117)
第 6 章 指针	(122)

6.1 地址、指针和指针变量的概念	(122)
6.2 指针的定义和用指针访问变量	(126)
6.3 指针的运算	(128)
6.4 用指针访问一维数组	(131)
6.5 用指针访问二维数组	(132)
6.6 用指针处理字符串	(137)
6.7 多级指针	(142)
6.8 用指针进行内存动态分配	(144)
习题六	(146)
第7章 C函数	(153)
7.1 模块化程序设计的思想	(153)
7.2 C函数的定义和调用	(154)
7.3 调用函数和被调用函数之间的数据传递	(159)
7.4 存储类型对函数调用的影响	(171)
7.5 函数的递归调用	(174)
7.6 main()函数的参数和返回值	(177)
习题七	(179)
第8章 复合数据类型	(189)
8.1 结构类型	(189)
8.2 联合类型	(208)
8.3 位段结构类型	(212)
8.4 枚举类型	(216)
8.5 类型定义 typedef	(217)
习题八	(219)
第9章 文件	(226)
9.1 文件概述	(226)
9.2 文件的打开与关闭	(228)
9.3 文件的读写操作	(230)
9.4 文件检测函数	(237)
9.5 文件的顺序存取和随机存取	(239)
习题九	(247)
第10章 编译预处理和分割编译	(254)
10.1 编译预处理	(254)
10.2 分割编译	(262)
习题十	(265)
附录	(270)
附录1 C语言运算符集	(270)
附录2 ASCII代码表	(271)
附录3 Turbo C 2.0 常用库函数及其标题文件	(272)
附录4 Turbo C 2.0 编译错误信息	(276)
附录5 习题参考答案	(279)

第 1 章 C 程序设计的初步知识

C 语言是一种面向过程的通用程序设计语言。它以表达简明、使用灵活、结构化的流程控制、丰富的数据结构和操作符集合、良好的程序可移植性和高效率的目标代码为特征。C 语言不仅具有高级语言的要素,还兼有低级语言的功能,因此既可用于编写系统程序,也可用于编写不同领域的应用程序。本章主要介绍 C 程序的基本结构和书写风格、简单 C 程序的编制及 Turbo C 2.0 集成环境的使用方法。

1.1 C 语言的产生和发展

C 语言的祖先是 ALGOL60(ALGOriThm Language)。ALGOL60 是 1960 年由国际计算机委员会设计的一种面向过程的结构化程序设计语言,用它编写的程序具有可读性和可移植性好的特点。但是,它不能直接对硬件进行操作,不宜用来编写系统程序。系统程序主要用汇编语言编写,而汇编语言是面向机器的,用它编写的程序可读性和可移植性都比较差。为此,人们开始考虑设计一种集高级语言和低级语言功能于一身的语言,以使用它来编写可读性和可移植性都比较好的系统程序。

1963 年,英国的剑桥大学和伦敦大学首先将 ALGOL60 发展成 CPL(Combined Programming Language)。该语言已比较接近于硬件,但规模较大,难以实用。

1967 年,剑桥大学的 Martin Richards 将 CPL 改制成 BCPL(Basic Combined Programming Language)。BCPL 比 CPL 大为简化,既具有结构化程序设计语言的特点,也能直接处理与硬件相关的数据,被软件人员用做系统程序的描述语言。

1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 修改成 B 语言(Boiling CPL down to its basic good feature),并用 B 语言开发了第一个高级语言 UNIX 操作系统,在 DEC 公司的 PDP-7 小型机上运行。

1972 年,Dennis M. Richie 将 B 语言修改设计成 C 语言。C 语言既保持了 BCPL 和 B 语言的精炼和接近于硬件的特点,也克服了它们过于简单、数据无类型等缺点。1973 年,Ken Thompson 和 Dennis M. Richie 又合作将 1969 年用汇编语言编写的 UNIX 操作系统改用 C 语言编写,其中 C 语言代码占 90% 以上,只保留了少量汇编语言代码。这样就使得 UNIX 操作系统向其他类型的机器上移植变得相当简单。到了 20 世纪 70 年代中期,UNIX 操作系统和 C 语言作为软件设计师的良好工具传遍了贝尔实验室,接着也传遍了几乎所有的美国大学校园。随着西欧和日本相继宣布加入 UNIX 和 C 的行列,UNIX 和 C 开始风靡世界。

1978 年,以 UNIX 第 7 版中的 C 编译程序为基础,Brain W. Kernighan 和 Dennis M. Richie 合著了影响深远的名著《The C Programming Language》。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础,称为 K&R C 语言。其后的十几年中,适用于不同机种和不同操作系统的 C 编译系统相继问世,从而把 C 语言的应用推向了更加广泛普及的阶段。1983 年美国国家标准局 ANSI 制定了 C 语言标准。这个标准不断完善,并从 1987 年开始实施,称

为 ANSI C。1988 年, Kernighan 和 Richie 修改了经典著作《The C Programming Language》, 按 ANSI C 标准重新编写了该书。现在一般称 ANSI C 为新标准或现代 C, K&R C 为旧标准或传统 C。此后陆续出现的各种 C 语言版本, 如 Microsoft C 5.0、9.0, Turbo C 2.0、3.0, Quick C 等都是与 ANSI C 兼容的版本。它们的语法和语句功能是一致的, 差异表现在各自的标准函数库中收纳的函数种类、格式和功能上, 尤其是图形函数库的差异更大一些。

C 语言是一种面向过程的语言, 意思是用 C 语言编程时, 必须按照算法的实现过程逐条语句编写, 通知计算机一步一步怎么做。进入 20 世纪 80 年代后, 面向对象的程序设计概念日益普及。所谓面向对象, 是指通过类和对象把程序所涉及的数据结构和对它施行的操作有机地组织成模块, 对于数据和对数据的处理细节进行最大限度的封装, 从而使开发出来的软件易重用、易修改、易测试、易维护、易扩充。正如其他传统的程序设计语言都在发展自己面向对象的新版本一样, C 语言也在发展的同时, 朝着支持面向对象程序设计(Object Oriented Programming)的方向迈出了步伐。1986 年, 美国 AT&T 的贝尔实验室的 Bjarne Stroustrup 推出了 C 语言的超集 C++ 语言, 也叫“带类的 C”。在软件开发领域, C 和 C++ 一直是最有生命力的程序设计语言, 大多数优秀的程序员都是 C 和 C++ 语言的高手, 大多数成功的软件也都是用 C 或 C++ 语言编制的。

随着信息时代的到来, 由于设计 Internet 上的 Web 浏览器的需要, 1994 年出现了 Java 语言。它不仅支持 OOP, 而且具有软硬件平台无关性的特点, 更适合于程序员进行网络开发。Java 脱胎于 C++, 被称为 C++ 的衍生语言。2000 年, Microsoft 推出了其下一代计算计划——Microsoft .NET。它是一个具有公共语言子集的开发平台, 实现了多种语言及其类库的无缝集成, 使应用程序的开发更容易、更简单, 并可能成为下一代网络通信标准。C# 是专为这一平台推出的全新语言, 它也派生于 C 和 C++, 并具有语法简洁、面向对象、与 Web 紧密结合、有卓越的安全性及灵活性和兼容性俱佳等特点, 成为 .NET 平台一流的网络编程工具。

1.2. C 程序的基本结构和书写风格

C 程序的基本结构是函数, 并采用完全自由的书写风格。

1.2.1 一个简单的 C 程序

【例 1.1】给定圆的半径 r , 计算圆的周长 c 和面积 a 。

```
# define PI 3.14159 /* 宏定义 */
main() /* 函数名 */
{ /* 函数体开始 */
    float r, a, c /* r:半径;a:面积;c:周长 */
    r=2.5; /* 给定 r 的值 */
    a=PI*r*r; /* 计算 a */
    c=2*PI*r /* 计算 c */
    printf("r= %f, a= %f, c= %f \n", r, a, c); /* 输出 r, a, c */
} /* 函数体结束 */
```

程序根据给定的半径 $r=2.5$ 计算面积 a 和周长 c , 并在屏幕上显示 r, a, c 的值。其中,

PI 代表圆周率,用 # define 定义成符号常数;函数名有固定的名称 main();用花括号括住的部分称为函数体,是程序的定义和执行部分;a、r、c 被定义为浮点型(即实数)变量,分别代表半径、面积和周长;printf()是 C 编译系统提供的输出函数,用来输出结果;在“/*”和“*/”之间的文字称为注释。程序运行结果如下:

```
r = 2.500000, a = 19.634956, c = 15.707965
```

1.2.2 C 程序的基本结构

C 程序的基本结构是函数,一个或多个 C 函数组成一个 C 程序,若干 C 语句构成一个 C 函数,若干基本单词形成一个 C 语句。

1. C 函数

C 函数是完成某个整体功能的最小单位,是相对独立的模块。一个简单的 C 程序可能只有一个主函数,而复杂的 C 程序则可能包含一个主函数和任意多个其他函数,但所有的 C 函数都具有相同的结构。

(1) 函数名

主函数有固定的名称 main,其他函数则可以根据标识符的命名方法任意取名。主函数通常包括整个程序的轮廓,由它再调用其他函数。

(2) 形式参数

在函数名的后面有一对圆括号,其中放置一个或多个形式参数,简称形参、虚参或哑元。一个函数也可以没有形式参数,但圆括号不能省略。

(3) 函数体

用花括号包围起来的部分是函数体,即函数的主体。它主要有两大部分:第一部分是本函数内部用到的局部变量类型定义(C 语言中,所有的变量都要先定义后使用);第二部分是语句序列,完成本函数的功能。

图 1.1 描述了 C 程序的一般格式,除了 main() 函数外,函数 fun1() 到 funn() 均为用户自行命名的函数。程序执行时,无论各个函数的书写位置如何,总是先执行 main() 函数,由 main() 函数调用其他函数,最后终止于 main() 函数(有关函数的详细介绍见第 7 章)。

2. C 语句

C 语句是完成某种程序功能(如赋值、输入、输出等)的最小单位,所有的 C 语句都以分号结尾。C 语句可分为表达式语句、复合语句和空语句三类。

(1) 表达式语句

任何 C 表达式末尾加上分号后,就构成一条表达式语句。如

```
i = 0;
x = x + 1;
```

等。

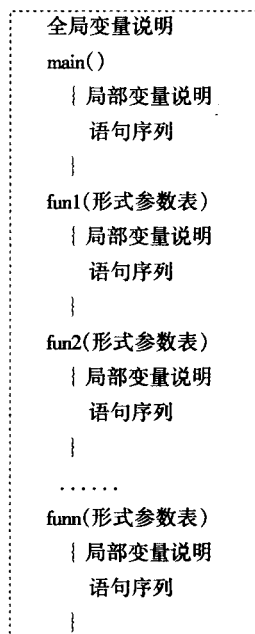


图 1.1 C 程序的一般格式

(2)复合语句

一组 C 语句用花括号括住,就构成复合语句。如

```
while(i < 10)
{
    sum = sum + i;
    i + +;
}
```

复合语句被视为一个整体,通常用在条件分支或循环语句中。有时为了数据隐藏的目的,用复合语句形成一个代码块,块中定义的局部变量不会对程序的其他部分发生副作用。

(3)空语句

只有一个分号的语句称为空语句。如

```
for(i = 0; i < 1000; i + +)
;
```

由一条 for 语句(循环语句)和一条空语句组成。空语句用作循环语句的循环体,表示什么也不做。事实上,这个循环的功能是延迟一小段时间。有时,空语句被用作转向点。

3. 基本单词

一个 C 语句由若干个基本单词组成。C 语言共有五种基本单词,即关键字(亦称保留字)、标识符、常数、操作符和分隔符。例如语句

```
float r, a, c;
```

中, float 是关键字(代表数据类型), r、a 和 c 是标识符(表示变量);又如语句

```
a = PI * r * r, c = 2 * PI * r;
```

中,“=”、“*”和“,”是操作符,2 是常数,PI 是符号常数。

(1)关键字

关键字是 C 语言中有特定意义和用途、不得作为他用的字符序列,其中 ANSI C 标准规定的关键字有 32 个, Turbo C 扩充的关键字有 11 个,见表 1.1。所有的 C 关键字都必须小写。

表 1.1 Turbo C 关键字一览表

ANSI C 标准关键字					
auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	int	long
register	return	signed	short	sizeof	static
struct	switch	typedef	union	unsigned	void
volatile	while				
Turbo C 扩充关键字					
asm	interrupt	near	far	_ds	pascal
_ss	cdecl	_cs	huge	_es	

(2)标识符

标识符用来表示变量名、数组名、函数名、指针名、结构名、联合名、枚举常数名、用户定

义的数据类型名及语句标号等用途的字符序列,可由1~32个字符组成,第一个字符必须是字母或下画线,后面的字符可以是字母、数字或下画线。例如

AB, Ab, aB, ab, A_b, _ab, ab_, s2d, W_length

等都是正确的标识符,而

A + B, A'B, A.B, 2abc, α , β , d%

等是错误的标识符。

标识符不能与C关键字相同,而且区分大小写。例如AB和Ab是两个不同的标识符,ELSE可以作为标识符,它不会与C关键字else混淆。

顺便指出,在C编译系统的库函数中,经常使用以下画线“_”打头的函数名或变量名,所以在程序中也应尽量避免使用以“_”打头的标识符,免得与库函数冲突。

(3) 常数

常数包括数值常数(如123、-23.5、1.2E4等)、字符常数(如'a'、'B'、'c'等)、字符串常数(如"xyz"、"good morning"等)、符号常数以及枚举常数(第8章介绍)。

(4) 操作符

操作符包括各种运算符(如+、-、*、/),有特定意义的标点符号(如花括号、方括号、圆括号、逗号)等。

(5) 分隔符

分隔符用来分隔相邻的标识符、关键字和常数,最常用的分隔符是空格,此外还可以用制表符、换行符、换页符等作为分隔符。

1.2.3 C程序的书写风格

1) 函数 在整个程序文件中函数可以出现在任意位置。主函数不一定出现在程序的开始处,但不管主函数位于程序中的何处,程序运行时总是从主函数开始。

2) 语句数量 每个程序行中的语句数量任意,既允许一行内写几条语句,也允许一条语句分几行书写,但每条语句都必须以分号(;)结束。有时还可以在程序的适当地方(如两个函数之间)加进一个或多个空行,使程序结构更加清晰。

3) 注释 注释的位置任意,注释可以出现在程序的任何地方,既可以独占一行或几行,也可以出现在某语句的开头或结尾处。如果注释占有几行,则每一行都要以“/*”开始,以“*/”结束,“*”和“/”之间不能有空格。注释不是C语句,它对程序的编译和运行没有影响,使用注释的惟一目的是增加程序的可读性。

4) 书写格式 尽管C程序的书写几乎没有限制,但为使程序清晰易读,通常每行写一条语句;不同结构层次的语句从不同的位置开始,即按缩进格式书写成阶梯形状,可以用Tab键或空格键调整各行的起始位置。

1.3 简单的C程序设计

一个程序通常具备三个功能,即输入数据、数据运算和输出结果。在C语言中,数据运算主要是由赋值语句完成的,数据的输入、输出则需要调用C编译系统提供的输入、输出函数。本节简要介绍赋值语句和格式输入、输出函数的使用,以便能进行最简单的程序设计。

1.3.1 赋值语句的简单使用

赋值语句具有计算和存储两大功能,一般格式为

```
v = e;
```

其中,v是变量名;e是表达式。例如, $x = a + b$,其功能是先计算表达式 $a + b$ 的值,并把该值转换成和 x 相同的数据类型后保存在变量 x 中。

1.3.2 格式输入输出函数的简单使用

输入指的是通过输入设备将原始数据送入计算机,以便计算机对它们进行处理;输出是将保存在内存中的计算结果送到输出设备上,以便阅读。由于 C 语言不提供输入输出语句,C 程序中的输入和输出主要是通过 C 编译系统提供的输入、输出函数实现的。其中用得最多的是格式输出函数 `printf()`和格式输入函数 `scanf()`。本节只对它们进行简单介绍,详细介绍见第 2.5 节。

1. 格式输出函数 `printf()`

格式输出函数 `printf()`用来按指定的格式输出数据,是内存与显示器之间进行数据交换的主要手段,也是用户获得程序运行结果的主要途径。其一般调用形式为

```
printf("格式控制字符串",输出项目清单);
```

其中,`printf`是函数名,其后的括号中是该函数的参数;格式控制字符串用双引号括住,用来规定输出格式,例如,`%d`用来输出十进制整数,`%f`用来输出实数;输出项目清单中包含零个或多个输出项,它们可以是常数、变量或表达式,当有多个输出项时,相互之间用逗号隔开,例如

```
printf("%f, %f", x, y);
```

用来按小数形式输出变量 x 和 y 的值。

2. 格式输入函数 `scanf()`

格式输入函数 `scanf()`用来按指定的格式接收输入数据,是内存与键盘之间进行数据交换的主要手段,也是用户为程序运行提供原始数据的主要途径。其一般调用形式为

```
scanf("格式控制字符串",输入项目清单);
```

其中,`scanf`是函数名,其后的括号中是该函数的参数;格式控制字符串用双引号括住,用来规定输入格式,其用法和 `printf()`相同;输入项目清单中至少包含一个输入项,且必须是变量的地址(变量地址的表示形式是在变量名前面加一个“&”),当有多个输入项时,相互之间用逗号隔开,例如

```
scanf("%f%d", &a, &b);
```

用来接收从键盘输入的一个实数和一个整数,并分别存放在变量 a 和 b 中。

1.3.3 库函数和标题文件

上面介绍的 `printf()`和 `scanf()`是 Turbo C 编译系统提供的库函数。库函数不是 C 语言的组成部分,而是由 C 编译系统提供的一些非常有用的功能函数,例如各种输入输出函数、数学函数、字符串处理函数等,可供用户在自己的程序中直接调用。这些库函数的说明、类型和宏定义都分门别类地保存在相应的标题文件(也称头文件)中,而对应的子程序则存放在运行库(.lib)中。当使用系统提供的库函数时,只要在程序开始用

```
# include <标题文件>
```

或

```
# include "标题文件"
```

就可以调用其中定义的库函数。printf()和 scanf()是在标题文件 stdio.h 中定义的,因此在调用它们之前,应先用

```
# include <stdio.h>
```

或

```
# include "stdio.h"
```

将它们所在的标题文件包含进来。这里,由 # 开头的 include 行以及前面用到的 # define 都称为命令行。它们是 C 语言的编译预处理命令(详见第 10 章),不是 C 语句,命令动词 include 和 define 也不是 C 关键字。

Turbo C 提供的常用标题文件如下(标题文件名不区分大小写):

ALLOC.H	动态分配函数
CONIO.H	屏幕处理函数
CTYPE.H	字符处理函数
GRAPHICS.H	图形函数
MATH.H	数学函数
STDIO.H	标准输入输出及文件操作函数
STDLIB.H	标准实用函数
STRING.H	字符串处理函数

从技术角度讲,任何函数都可以自行编制,完全可以不使用 C 编译系统提供的函数库。然而人们很少这样做,因为函数库 C 编译系统提供的一种重要的软件资源,充分利用它们,避免重复自行编制,可达到事半功倍的效果;另一方面,C 语言为了减少对硬件的依赖,不提供任何执行 I/O 操作的方法。这时,自行编制 I/O 函数也存在很大困难。读者应在学习 C 语言本身的同时,逐步了解和掌握各种库函数的功能和用法,以简化编程。

需要说明的是,不同的 C 编译系统提供的库函数在数量、种类、名称及使用上都有一些差异,甚至连标题文件的名称也不一定相同。

1.3.4 简单程序设计举例

【例 1.2】计算任意立方体的表面积和体积。

设一个立方体的三条边分别为 a、b、c。若从键盘输入这三条边长,就可以计算任意立方体的表面积和体积。程序如下:

```
# include <stdio.h>
main()
{ float a,b,c,s,v;
  scanf("%f%f%f",&a,&b,&c); /* 输入三条边 */
  s=2*(a*b+b*c+a*c); /* 计算立方体表面积 s */
  v=a*b*c; /* 计算立方体体积 v */
  printf("a= %f b= %f c= %f \n",a,b,c); /* 输出边长 */
  printf("s= %f \n",s); /* 输出表面积 */
  printf("v= %f \n",v); /* 输出体积 */
}
```

程序运行时,从键盘输入 1、2 和 3,其间用空格分隔,并以回车键结束,屏幕显示如下:

```
1 2 3 ↓  
a = 1.000000  b = 2.000000  c = 3.000000  
s = 22.000000  
v = 6.000000
```

可以多次运行该程序,每次输入不同的边长,就可以输出不同的计算结果。

【例 1.3】从键盘输入一个小写字母,打印该字母及其对应的十进制 ASCII 代码值,然后打印该字母对应的大写字母及其对应的十进制 ASCII 代码值。

已经知道,同一个英文字母的大小写字母的十进制 ASCII 代码值相差 32,于是有了小写字母就容易得到大写字母,反之也一样。程序可编制如下:

```
main()  
{ char ch1, ch2;  
  scanf("%c", &ch1);  
  printf("%c, %d \n", ch1, ch1);  
  ch2 = ch1 - 32;  
  printf("%c, %d \n", ch2, ch2);  
}
```

程序运行后,从键盘输入任意一个小写字母,屏幕先显示该小写字母及其对应的十进制 ASCII 代码值,然后显示对应大写字母及其对应的十进制 ASCII 代码值。某次运行结果为:

```
d ↓  
d, 100  
D, 68
```

1.4 C 程序的开发过程

C 语言是一种编译型的程序设计语言。一个 C 程序要经过编辑、编译、连接和运行四个步骤,才能得到运行结果。

1. 编辑

编辑是指输入 C 语言源程序和进行修改,最后以文本文件的形式存放在磁盘上,文件名由用户自己选定,扩展名一般为“.c”,例如 a.c、sample.c、gfc.c 等。

2. 编译

编译是把 C 源程序翻译成可重定位的二进制目标程序。编译过程由编译程序完成。编译程序自动对源程序进行句法和语法检查。当发现错误时,就将错误类型和错误在程序中的位置显示出来,以帮助用户修改源程序中的错误。如果未发现句法和语法错误,就自动形成目标代码并对目标代码进行优化后生成目标文件。目标文件的名称由编译系统自动规定,也可以由用户指定。

3. 连接

连接也称链接或装配,是用连接程序将编译过的目标程序和程序中用到的库函数连接装配在一起,形成可执行的目标程序。可执行目标文件的扩展名由系统自动确定。

4. 运行

运行是将可执行的目标文件投入运行,以获取程序的运行结果。通常,在 DOS 操作系统提示符下直接键入可执行文件名,或在 Windows 操作系统下用鼠标双击可执行文件名即可。

上述四个步骤示于图 1.2。图中,带箭头的实线表示操作流程,带箭头的虚线表示操作所需要的条件和产生的结果。例如,在 Turbo 2.0 集成环境中的操作步骤为:在编辑菜单下进行输入和编辑,产生 C 源程序文件(扩展名为 .c);对该文件进行编译,生成可重定位的目标文件(扩展名为 .obj);再将目标文件进行连接装配,生成可执行的目标文件(扩展名为 .exe),也称可执行文件;运行可执行文件即可获得运行结果。

1.5 Turbo C 2.0 集成环境的使用

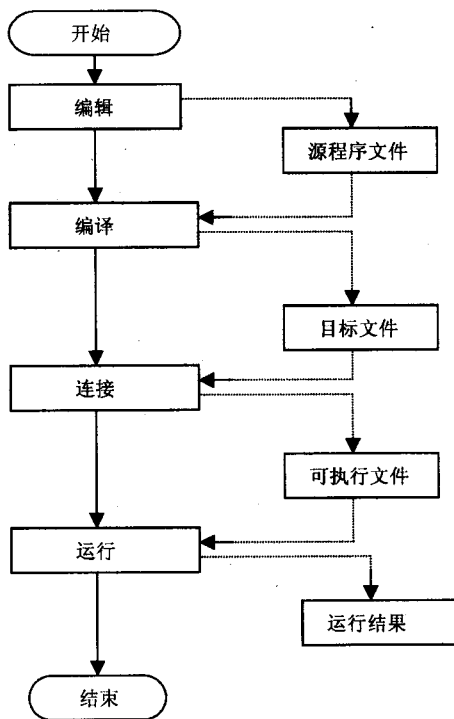


图 1.2 C 程序的开发过程

Turbo C 2.0 是美国 Borland 公司开发的一个 C 语言集成环境,它集编辑、编译、连接及运行功能于一身,使得 C 程序的编辑、调试和测试非常简捷,编译和连接速度极快,使用也很方便。

1. Turbo C 2.0 的启动

Turbo C 2.0 可以在 DOS 或 Windows 环境下运行。

(1) DOS 环境下启动

假设 Turbo C 2.0 安装在 C 盘根目录下的 TC 子目录中,要启动 Turbo C,只要先启动 DOS。如果需要使用汉字,则再启动汉字 DOS(目前,只有在 UC DOS 或天汇 DOS 下, Turbo C 才能直接处理汉字),然后可以按下面两种方式进入 Turbo C 集成环境。

① 直接进入 TC 子目录,启动 Turbo C,即

C> CD C: \ TC ↓

C: > tc ↓

② 先进入用户子目录(假设用户子目录为 C: \ USER),然后启动 Turbo C,即

C> CD C: \ USER ↓

C: \ USER > C: \ TC \ tc ↓

使用前一种方式进入 Turbo C,用户的程序文件将存放在 TC 子目录下;使用后者,用户的程序文件将存放在用户子目录下。

(2) Windows 环境下启动

在 Windows 环境下,打开“资源管理器”,找到 Turbo C 所在的文件夹,用鼠标左键双击该文件夹下的 tc.exe 文件。