



普通高等教育“十一五”国家级规划教材
21世纪高职高专规划教材 (计算机类)

C 语言程序设计

刘莹 主编



机械工业出版社
CHINA MACHINE PRESS



配多媒体课件





清华大学出版社
清华大学出版社
清华大学出版社

C 语言程序设计



清华大学出版社



清华大学出版社

普通高等教育“十一五”国家级规划教材
21世纪高职高专规划教材（计算机类）

C 语言程序设计

主 编 刘 莹
副主编 楼建忠
参 编 胡忭利 张爱玲 孙 雪



机械工业出版社

本书为普通高等教育“十一五”国家级规划教材，是以 C 语言初学者为主要对象的程序设计教程。

本书共分 11 章。其中包括 C 语言基础知识、顺序结构程序设计、分支结构程序设计、循环结构程序设计、数组、指针、函数、用户标识符的作用域和存储类别、结构体与其他构造类型、文件、实例解析。大部分教学内容采用例题的形式进行组织，整体结构编排合理，组织形式新颖，例题丰富，符合学生的认知规律和学习特点。通过本书的学习，使学生能够掌握程序设计的基本思想和常见简单问题的算法，并可以编写程序加以实现。本书还提供了贯穿于整个教学过程的综合实例。实例涉及本书的全部教学内容，学生可以随着讲授内容的不断增多，逐步补充和完善其程序的功能，从而在教学的整个过程中，培养了学生对应用程序的开发和设计能力。

本书可作为高等院校学生学习 C 语言的教材，也可作为 C 语言自学者的参考书。

本书配有电子教案，凡一次性购书 30 本以上者免费赠送一份电子教案。请与本书策划编辑余茂祚联系（联系电话 010—88379759，邮箱 yumaozuo@163.com）。

图书在版编目（CIP）数据

C 语言程序设计/刘莹主编. —北京：机械工业出版社，2008.9
普通高等教育“十一五”国家级规划教材. 21 世纪高职高专规划教材
(计算机类)

ISBN 978-7-111-25069-2

I. C… II. 刘… 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（2008）第 137816 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：余茂祚 责任编辑：余茂祚 吴超莉 版式设计：张世琴
责任校对：刘志文 封面设计：赵颖哲 责任印制：李妍

北京鑫海金澳胶印有限公司印刷装订

2009 年 1 月第 1 版第 1 次印刷

184mm × 260mm · 16.5 印张 · 409 千字

0001—4000 册

标准书号：ISBN 978-7-111-25069-2

定价：27.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

销售服务热线电话：（010）68326294

购书热线电话：（010）88379639 88379641 88379643

编辑热线电话：（010）88379759

封面无防伪标均为盗版

21 世纪高职高专规划教材

编委会名单

编委会主任 王文斌

编委会副主任 (按姓氏笔画为序)

王建明	王明耀	王胜利	王寅仓	王锡铭	刘 义
刘晶磷	刘锡奇	杜建根	李向东	李兴旺	李居参
李麟书	杨国祥	余党军	张建华	茆有柏	秦建华
唐汝元	谈向群	符宁平	蒋国良	薛世山	储克森

编委会委员 (按姓氏笔画为序, 黑体字为常务编委)

王若明	田建敏	成运花	曲昭仲	朱 强	刘 莹
刘学应	许 展	严安云	李连邨	李学锋	李选芒
李超群	杨 飒	杨群祥	杨翠明	吴 锐	何志祥
何宝文	余元冠	沈国良	张 波	张 锋	张福臣
陈月波	陈向平	陈江伟	武友德	林 钢	周国良
宗序炎	赵建武	恽达明	俞庆生	晏初宏	倪依纯
徐炳亭	徐铮颖	韩学军	崔 平	崔景茂	焦 斌

总策划 余茂祚

前 言

本书融汇了作者多年的教学实践和项目开发的经验，不仅是一本集程序设计思想与具体实践为一体的教材，而且还兼顾了C语言程序设计的特点和学生学习的认知规律，改变了传统课程的“提出概念—解释概念—举例说明”的三部曲教学模式。编者经过不断探索和实践，总结出了便于学生学习和掌握的“提出问题—介绍解决问题的方法—归纳出一般规律”的新三部曲，并特别强调了通过例题提出问题的案例教学方法。教学实践证明，这种从抽象到具体，从个别到一般，从零散到系统的方法对学生学习程序设计方法是非常适合的。

本书除在章节的安排上做到了由浅入深、循序渐进外，还具备了以下特点：

1. 在遵循C语言程序设计教学体系的情况下，作者将大部分教学内容用例题的形式进行组织。根据要介绍的知识点精心组织、编写相应的例题，使学生的学习从阅读大量、正确、规范的程序开始，并在学习例题的过程中，掌握基本概念、语法规则、正确设计和编写程序。真正做到在解决实际问题中讲授语法。

2. 为了使学生对应用程序的设计、编写过程有一个全方位的了解和掌握，在本书中提供了一个贯穿于整个教学过程的实例。该实例涉及到本课程的全部教学内容，学生可以随着讲授内容的不断深入，逐步补充和完善其程序的功能，从而在教学的整个过程中，培养了学生对应用程序的开发、设计的能力，而不是局限于仅仅掌握相对独立的知识点。

3. 指针和函数是C语言的重点和难点。为了使能够顺利地接受新概念，在编写本书的过程中，作者尽量将有关算法安排在这两章之前，避免学生在接受指针和函数的同时，又要理解新的算法。这样的安排不仅容易突破难点，而且有利于知识的巩固与提高。

本书共分11章。第1、10章由西安理工大学高等技术学院的胡忭利编写，第2、8章由西安理工大学高等技术学院的张爱玲编写，第3~6章及附录由北京联合大学的刘莹编写，第7、9章由金华职业技术学院信息学院的楼建忠编写，第11章由北京联合大学自动化学院的孙雪编写。

目录中标注“*”的章节可作为选学内容。

本书由刘莹任主编并统稿，楼建忠任副主编。孙雪负责全书的版面编排工作。

由于编者水平有限，书中不足和欠妥之处在所难免，恳请广大读者批评指正。

编 者

目 录

前言

第 1 章 C 语言基础知识 1

- 1.1 计算机语言概述 1
- 1.2 C 语言概述 3
- 1.3 C 语言程序的组成和格式 4
- 1.4 C 语言程序的开发过程及上机
步骤 8
- 1.5 数据类型 11
- 1.6 常量 12
- 1.7 变量 18
- 1.8 C 语言的运算符和表达式 21
- 1.9 不同类型数据间的混合运算和
转换 26

习 题 28

第 2 章 顺序结构程序设计 31

- 2.1 结构化程序设计的基础 31
- 2.2 赋值语句 35
- 2.3 字符的输入/输出 35
- 2.4 格式输入/输出 37

习 题 42

第 3 章 分支结构程序设计 46

- 3.1 关系运算符与关系表达式 47
- 3.2 逻辑运算符与逻辑表达式 48
- 3.3 if 语句 50
- 3.4 switch 语句 61

习 题 65

第 4 章 循环结构程序设计 67

- 4.1 for 语句 68
- 4.2 while 语句 73
- 4.3 do-while 语句 76
- 4.4 break 语句和 continue
语句 78
- 4.5 循环的嵌套 82

习 题 84

第 5 章 数组 88

- 5.1 一维数组 89
- 5.2 字符串与字符数组 103
- 5.3 二维数组 112

习 题 118

第 6 章 指针 120

- 6.1 指针与指针变量 120
- 6.2 指针变量的定义和引用 121
- 6.3 一维数组与指针变量 126
- 6.4 字符串与指针变量 130
- 6.5 * 指针与二维数组 133

习 题 135

第 7 章 函数 138

- 7.1 函数应用样例 138
- 7.2 函数定义和返回值 139
- 7.3 函数的调用 141
- 7.4 调用函数和被调用函数之间的
数据传递 145
- 7.5 函数的嵌套调用与递归
调用 155

习 题 159

第 8 章 用户标识符的作用域和存储 类别 161

- 8.1 局部变量和全局变量 161
- 8.2 局部变量的作用域和生
存期 163
- 8.3 全局变量的作用域和生
存期 166
- 8.4 函数的存储类别 169

习 题 170

第 9 章 结构体与其他构造类型 175

- 9.1 结构体类型变量的定义和

引用	175	第 11 章 实例解析	221
9.2 结构体类型数组与指针	179	11.1 系统的功能介绍	221
9.3 函数之间结构体类型的数据 传递	182	11.2 系统的功能实现	225
9.4 * 链表	186	11.3 系统测试	244
9.5 共用体	195	11.4 使用文件完善功能	249
9.6 枚举类型	197	附录	252
习 题	197	附录 A C 语言关键字	252
第 10 章 文件	200	附录 B 运算符的优先级和结合 方向	252
10.1 文件概述	200	附录 C 常用字符与 ASCII 代码 对照表	253
10.2 文件的基本操作	203	附录 D 常用 C 语言库函数	254
10.3 文件的读/写操作	207	参考文献	257
10.4 文件操作中的其他函数	213		
习 题	218		

第 1 章 C 语言基础知识

本章主要内容

- C 语言程序的基本概念和上机步骤
- C 语言的基本数据类型
- 常量与变量
- 算术运算符与赋值运算符

1.1 计算机语言概述

1.1.1 计算机语言

若要让计算机解决某个问题，必须先将该问题分解为一系列的解题步骤，然后将这些解题步骤用计算机语言表示为一组指令，计算机执行这组指令才能使问题得到解决。这组解决实际问题的指令，就是通常所说的程序，即用某种计算机能理解并执行的语言，把解题的方法和步骤描述出来。描述这组指令的语言就是编程语言。为计算机编排和调试程序的过程称为程序设计。程序设计要选择一种适当的编程语言，所以又将编程语言称为程序设计语言。

从计算机问世以来，计算机编程语言也伴随着计算机技术的发展而不断发展，大体上经历了以下几个过程。

(1) 机器语言：机器语言是最早产生和使用的编程语言，其每条指令都是由二进制代码 0 和 1 组成的一个序列。任何一种计算机，都有各自的一套指令系统，每一套指令系统都由一组指令组成。某机器的指令系统就构成了该机器的机器语言，不同的机器系统有不同的机器语言，所以这种二进制代码形式的语言是面向机器的。

用机器语言编写的程序，计算机可以直接执行，但难以阅读和理解，且不易查错，只有少数专业人员才能掌握，编写程序相当困难，质量难以保证。

(2) 汇编语言：随着计算机应用范围的逐渐扩大，在 20 世纪 50 年代中期，人们开始用一些助记符来代替二进制代码 0、1 的序列。这种用助记符表示的语言称为符号语言或汇编语言。

用汇编语言编写的程序，效率和质量都有所提高，但计算机无法直接执行，必须借助一种工具，将汇编语言程序翻译成机器语言程序，这种工具就称为汇编程序，翻译的过程称为汇编。

与机器语言相比，汇编语言由于引入了助记符，使程序比较直观，也易于阅读和理解，但和机器语言一样，它也是面向机器的。使用机器语言和汇编语言都需要对计算机的内部结构有较深入的了解，因此用它们编写程序不仅劳动强度很大，而且程序的通用性也较差。通常将这两种语言称为计算机低级语言。

(3) 计算机高级语言：随着计算机技术的发展，其应用范围也越来越广，因此对程序的需求量也越来越大。面向机器的编程语言已不能满足要求，到 20 世纪 50 年代中期出现了第

一种高级语言——Fortran 语言。高级语言用一种接近于人类自然语言和数学语言的方式来描述计算机的操作，便于阅读、理解，移植性和通用性较好，因而得到了迅速普及和发展。高级语言不再面向机器，而是面向解题的过程，因而又被称为算法语言或过程语言。

使用高级语言编写程序时，不再需要熟悉计算机内部的硬件结构，而是将主要精力放到算法的描述上。同样，用高级语言编写的源程序，计算机不能直接执行，必须通过翻译程序将其翻译成机器语言的目标程序才可运行。一般有两种翻译方式：解释方式和编译方式。

解释方式：边翻译边执行，即翻译一句执行一句，不产生目标程序。

编译方式：将高级语言编写的源程序全部翻译成机器语言目标程序，再将目标程序交给计算机执行，最后得出程序执行的结果。高级语言源程序的编译执行过程为

高级语言源程序 → 编译系统 → 机器语言程序

1.1.2 程序设计语言简介

程序设计语言的种类很多，如 Quick Basic、Fortran、Pascal、Visual Basic、Java、C、C++、SQL 等，可应用在不同的场合发挥各自的优势。

这些程序设计语言可从不同的角度进行分类。按执行方式的不同可分为编译执行的语言和解释执行的语言；按解题过程的不同可分为过程化语言和非过程化语言；按思维模式的不同又可分为面向过程的程序设计语言和面向对象的程序设计语言。

1. 过程化程序设计语言和非过程化程序设计语言 在过程化程序设计语言中，程序设计人员进行程序设计，必须要面向解题过程，他们不仅要解决“做什么”的问题，还要解决“怎么做”的问题，程序设计者必须详细地规定出计算机操作的每一个细节，如 C 语言、Quick Basic、Fortran、Pascal 等均属于过程化程序设计语言。

而在另一些程序设计语言中，程序设计人员只需告诉计算机“做什么”，而无须去构造“怎么做”的过程，计算机就能完成所要求的操作，这种语言称为非过程化程序设计语言，如数据库语言 SQL、FoxBASE、FoxPro 等均属于非过程化程序设计语言。

非过程化程序设计语言使程序设计更为简便，为更多的人使用计算机提供了方便。但它的运行效率及灵活性都远不如过程化程序设计语言，而且非过程化的程序设计语言也只是在某些领域（如数据库管理）使用。

2. 面向对象的程序设计语言 随着计算机应用的深入，程序的规模越来越大，于是在 20 世纪 80 年代，人们又提出了一种面向对象的程序设计理念。它不仅仅是一种程序设计语言，应该说是一种全新的思维方式。它以一种更接近于人类的一般思维方式去看待程序设计。在程序设计中引入对象、属性、方法和事件等概念，使程序设计变得更加简练、自然。近几年，面向对象的程序设计语言得到了迅速的发展和应用，如 Visual Basic、C++、Java、Delphi 等语言。

1.1.3 常用计算机高级语言简介

目前，用于程序设计的语言已有数百种，每种语言都有各自不同的特点，可适用于不同的场合。机器语言和汇编语言虽然编程繁杂，但由于它是面向机器的，容易实现对硬件的访问，所以，它广泛应用于自动控制领域。高级语言由于其自身的优势，在各种不同场合得到了广泛的应用。下面对几种应用较为广泛的高级语言的特点及使用范围进行简单的介绍。

1) Fortran 语言是世界上出现最早的高级语言，是结构化的程序设计语言。在数值处理领域得到了广泛的应用，主要用于科学、工程计算等。

2) Basic 语言原是为初学者学习程序设计研制的一种语言, 该语言在 PC 上得到了广泛的应用。目前已有多个版本, 如 Borland 公司的 Turbo Basic、Microsoft 公司的 QBASIC 等。它们不仅简单易学, 而且具有结构化程序设计的特点, 是 DOS 环境下的程序设计语言。

3) Pascal 语言是 20 世纪 70 年代初期研制出来的结构化程序设计语言, 主要应用于教学, 目前在科学、工程计算领域和系统程序设计方面也得到了广泛的应用。

4) Visual Basic 语言是 Microsoft 公司推出的一种面向对象的“可视化”开发工具语言。它为用户提供了一个可视化的开发环境, 具有图形设计工具、结构化的事件驱动编程模式, 简单易学、功能强大, 可用于开发 Windows 环境下的各种应用程序。目前流行的 Visual Basic 6.0 深受用户的喜爱。

5) C 语言最初是作为操作系统设计语言而研制的, UNIX 操作系统就是用 C 语言实现的。C 语言具有很强的功能而且十分灵活, 它支持复杂的数据结构, 可大量使用指针, 有丰富的运算符和数据处理操作符, 并具有类似汇编语言的特性 (可直接对硬件进行操作)。C 语言既有一般高级语言的特性, 又具有低级语言的特性, 集两者的优点于一身。因此, C 语言不仅用于操作系统的设计, 也可用于设计各类应用程序。目前, C 语言已成为应用最为广泛的几种计算机语言之一, 流行的 C 语言版本是 Borland 公司的 Turbo C 2.0。

6) C++ 语言是在 C 语言的基础上发展起来的, 完全兼容 C 语言, 已成为当今最受欢迎的面向对象的程序设计语言。因为它既融合了面向对象的方法, 又融入 C 语言的许多重要特性, 使 C 语言程序员不必放弃自己已经十分熟悉的 C 语言, 而只要补充学习 C++ 和 Microsoft Visual C++ 知识即可。

7) Java 语言也是一种面向对象的程序设计语言, 是后起之秀, 大有后来者居上之势。它的基本功能类似于 C++, 但作了许多重大修改, 是一个纯面向对象的高级语言。Java 语言最大的特点是其跨平台特性, 即一次编译, 可适用于任何平台。Java 语言主要应用在网络编程上。

8) 其他语言。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简单、更容易, 并可能成为下一代网络通信标准。C# 是专门为这一平台推出的全新语言。它也派生于 C 语言和 C++ 语言, 并具有语法简单、面向大对象、与 Web 紧密结合、卓越的灵活性和安全性极佳等特点, 成为 .NET 平台一流的网络编程工具。

总之, 计算机语言很多, 读者要根据自己的需要选择学习。一般来说, C 语言作为学习计算机语言的入门语言很有特色和代表性, 也应是很多计算机爱好者的首选语言。

1.2 C 语言概述

C 语言是一种面向过程的通用程序设计语言, 其语言表达简明、使用灵活, 具有结构化的流程控制、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等特征。C 语言不仅可用于编写系统程序, 在其他许多领域也得到了很广泛的应用。

1.2.1 C 语言简介

C 语言的发展有一个不断演变、扩充和改进的过程, 且与 UNIX 操作系统密切相关。C 语言第一次大显身手是在 1973 年。这一年, 美国贝尔实验室的 Ken Thompson 和 Dennis

Ritchie 利用 C 语言重新编写了 UNIX 操作系统, 使 C 语言和 UNIX 操作系统成为从微机到巨型机上都适用的、最流行的语言和操作系统之一。

目前流行的 C 语言版本都是以标准 C (ANSI C) 为基础的, 各种版本的基本部分相同, 个别地方稍有差异, 在 PC 上经常使用的有 Microsoft C 5.0、Turbo C 2.0 等版本。本书的例题和习题都是在 Turbo C 2.0 编译环境下运行的。

随着面向对象编程技术的发展, 又出现了 C++、Turbo C++、Visual C++ 等 C 语言版本。C++ 是 C 语言的延伸, 是在 C 语言的基础上, 增加了面向对象的程序设计功能。因此, 本书只介绍基本 C 语言, 目的是让读者在掌握 C 语言基本语法的基础上, 掌握程序设计的一般方法。

1.2.2 C 语言的特点

C 语言发展非常迅速, 已成为系统程序设计和工程应用程序设计广泛采用的程序设计语言。这主要是由于它具有以下几方面强大的功能。

1) C 语言是中级语言。中级语言意味着把高级语言的成分同汇编语言的功能结合起来了, 或者说把高级语言的基本结构同低级语言的实用性结合起来了。C 语言可以直接访问内存的物理地址, 对二进制数据进行位操作, 具有程序易编、易读、易查错、易修改的特点。

2) C 语言是结构化程序设计语言。首先, C 语言的主要结构成分是函数, 即函数是基本功能模块, 是完成程序功能的基本构件。函数在程序中可以被定义完成独立的任务, 独立地编译成目标代码, 这样便可实现程序的模块化。其次, 每个函数又都由 3 种基本结构——顺序结构、分支结构和循环结构构成的, 使程序层次清晰, 便于使用、维护和调试。

3) C 语言是以函数的形式提供给用户的。它提供给用户大量丰富的库函数, 包括输入/输出函数、数学函数、字符处理函数和图形函数等, 可供用户直接调用。

4) C 语言的数据类型和运算符十分丰富, 可以方便地描述各种算法和运算, 使程序设计更加简便。

5) C 语言生成的目标代码质量高, 效率只略低于汇编语言, 与其他高级语言相比, 执行效率高。

6) C 语言的程序可移植性好, 容易从一种类型的计算机系统移植到另一种类型的计算机系统。

综上所述, C 语言是一种功能很强的语言。但是, 它也有一些不足之处, 如 C 语言语法限制不严谨, 对熟练的程序员来说, 编程灵活, 但安全性低; 再如, 运算符丰富, 实现的功能强, 但较难记忆和掌握。因此, 学习、使用 C 语言时, 不妨先学基本部分, 先掌握基本语法, 用熟练后再学编程技巧和一些灵活的语法, 进而全面掌握 C 语言。

1.3 C 语言程序的组成和格式

C 语言程序由 C 语言的若干语句序列组成, C 语言程序的基本结构是函数。通常一个 C 语言程序包含一个或多个函数, 一个函数由若干条 C 语言语句构成。

为了说明 C 语言程序的结构特点, 先看以下几个 C 语言程序。这几个程序由易到难, 表现了 C 语言程序在组成结构上的特点。虽然有关内容还未介绍, 但可从这些例子中了解到一个 C 语言程序的基本组成和书写格式。

1.3.1 简单的 C 语言程序

【例 1.1】 在显示器上显示输出一行字符“Hello! We are studying C”。

程序代码如下：

```
main( )           /* 定义主函数 */
{   printf("Hello! We are studying C\n");
}
```

程序的运行结果如下：

Hello! We are studying C

程序说明：

这是一个最简单的 C 语言程序，用于在显示器上输出一行字符。程序中用到了一个主函数，即 main 函数，这是每个 C 语言程序都必须要有的一个函数，且 main 后的一对括号（）不能省略。将要实现的所有操作要写在函数体中，即一对花括号 {} 中。本例函数体只包含了一条语句，调用系统提供的 printf 函数实现输出。

【例 1.2】 求半径分别为 1、3、4 的 3 个圆的圆面积并将圆面积值输出。

程序代码如下：

```
#define PI 3.14159           /* 命令行。定义符号常量 PI 来代表 3.14159 */
main( )                     /* 定义主函数 */
{   int r1=1,r2,r3;         /* 定义 3 个整型变量,用于存放 3 个圆的半径 */
    float s1,s2,s3;        /* 定义 3 个实型变量,用于存放 3 个圆的面积 */
    r2=3;r3=4;             /* 给圆半径变量赋值 */
    s1= PI * r1 * r1;      /* 计算第 1 个圆的面积 */
    s2= PI * r2 * r2;      /* 计算第 2 个圆的面积 */
    s3= PI * r3 * r3;      /* 计算第 3 个圆的面积 */
    printf("\ns1 = %f,s2 = %f,s3 = %f\n", s1 , s2 , s3 ); /* 输出计算结果 */
}
```

程序的运行结果如下：

s1 = 3.141590,s2 = 28.274309,s3 = 50.265442

程序说明：

本程序用于求 3 个给定半径的圆的面积。由于多次用到圆周率，故在程序中定义了一个符号常量 PI 来代替 3.14159。由于半径是整数，故定义了 r1、r2、r3 3 个整型变量用于存放半径；由于计算出的圆的面积是实数，故定义了 s1、s2、s3 3 个实型变量用于存放圆的面积。另外，还可看到每个语句的最后都有一个分号“;”，这是 C 语言的语句结束符。程序中的 /* */ 是注释部分，用于对程序的注解说明，编译和运行时被系统忽略。

【例 1.3】 求任意给定的两个整数中的最大数并将其输出。

程序代码：

```
/* max 函数是用户自定义的函数,用于求两个整型数中的最大数 */
max( int x , int y)       /* 函数的声明部分 */
{   int z;                /* 定义一个整型变量 z */
    if (x>=y)             /* 比较 x 和 y,把大者给 z */
        z=x;
    else
```

```

        z=y;
    return( z );          /* 返回 z 的值到主调函数中 */
}
main( ) /* 主函数完成两个整数的输入,调用 max 函数,输出两个数中的最大数 */
{
    int a , b , large ;
    scanf("%d, %d" , &a, &b);      /* 从键盘输入两个整数分别给变量 a 和 b */
    large = max( a , b );          /* 调用 max 函数,求 a,b 中的最大数 */
    printf("\n large = %d\n",large); /* 输出计算结果 */
}

```

程序的运行结果如下:

```
6,23<Enter>
```

```
large = 23
```

程序说明:

本程序由两个函数组成: main 函数和 max 函数。main 函数用于实现两个整数的输入,并负责调用 max 函数,最后输出 max 函数的返回值(两个数中的最大数)。main 函数可以放在程序最前面,也可以放在程序最后,本程序中 main 函数放在程序最后。max 函数是用户自定义的函数,用于求两个整数中的最大数,它是通过 main 函数来调用的, max 函数通过 return 语句将值返回给了 main 函数并将值赋给了 large 变量,最后在 main 函数中输出运算结果。

1.3.2 C 语言程序的结构特点

从上述 3 个例子,我们可以总结出 C 语言程序的结构有如下特点。

1) C 语言程序是由函数构成的,可以有一个或多个函数,但其中必须有且仅有一个 main 函数。main 函数的位置任意,可在用户自定义函数的前面,也可在它的后面。无论 main 函数放在任何位置,C 语言程序总是从 main 函数开始执行。

2) 每个函数由两部分组成,即函数头(也称为函数的说明部分)和函数体。其结构如下:

函数类型 函数名(形参表) —— 函数头

```

{
    说明部分          }
    可执行部分      } 函数体
}

```

注意:函数体部分要写在一对花括号 {} 内。

3) C 语言程序中的每个语句,都以分号“;”作为语句结束符,即“;”是一个语句的结束标志。

4) C 语言程序的书写格式自由。可以一行写一条语句,也可以一行写多条语句,还有一条语句分几行写。通常,一行写一条语句,便于阅读。

5) 可以用 /* …… */ 对程序的任何部分进行注释。注释内容要写在一对“/* */”符号之间。注释部分允许出现在程序中的任何位置上。

6) C 语言程序本身没有输入/输出语句。它的输入/输出操作由 scanf 和 printf 库函数完成。

7) 主函数可以调用任何其他函数,任何其他函数之间也可以相互调用,但不能调用主

函数。

1.3.3 书写 C 语言程序时应遵循的规则

从书写清晰，便于阅读、理解和维护的角度出发，在书写 C 语言程序时应遵循以下规则：

- 1) 一个说明或一个语句占一行。
- 2) 用花括号 {} 括起来的部分，通常表示了程序的某一层结构。

3) 低一层次的语句或说明可比高一层次的语句或说明向右缩进若干个格后书写，以便看起来结构更加清晰，增加程序的可读性。用户在编程时应力求遵循这些规则，以养成良好的编程习惯。

4) 标识符、关键字之间至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。但函数名与其后面的括号之间不能加空格。

1.3.4 C 语言程序的基本词法

任何一种语言，都有自己的符号、单字、单词以及语句的构成规则。C 语言作为计算机的一种程序设计语言，也有自己的字符集、基本词类（关键字或保留字）、标识符及其命名规则。只有学习、遵从它们，才能编写出符合要求的各种程序来。

1. 字符集 允许出现在 C 语言程序中的所有字符的总体，称为 C 语言的字符集。它是构成程序设计语言的最小语法单位。C 语言的基本字符集包括数字、英文字母、特殊符号及转义字符。

(1) 数字：10 个十进制的数字，即 1, 2, 3, 4, 5, 6, 7, 8, 9, 0。

(2) 英文字母：26 个大写英文字母 A~Z, 26 个小写英文字母 a~z。

(3) 特殊符号：空格, !, #, %, &, *, (,), +, -, _, =, <, >, /, |, \, {, }, [,], ,, ,;, :, ', ", ~, @, \$, ^, ? 等。

(4) 转义字符：在 C 语言程序中，可以用转义字符来表示键盘上的字符以及某些不可见的功能控制符。有关转义字符的内容将在后面介绍。

2. 关键字 在 C 语言中，把那些具有特定含义和用途且不得作为它用的字符序列称为关键字或保留字。这些关键字用于构成 C 语言中语句的成分或作为存储类型和数据类型说明，而不能作为其他用途。C 语言的关键字有 32 个，只能小写，见表 1-1。随着学习的深入，这些关键字以后慢慢都会遇到。

表 1-1 C 语言的关键字

保留字	含义	保留字	含义	保留字	含义
char	字符型	void	空值型	while	当
int	整型	const	常量型	do	做
long	长整型	volatile	可变量型	break	终止
short	短整型	auto	自动	continue	继续
float	单精度实型	extern	外部	goto	转向
double	双精度实型	static	静态	return	返回
unsigned	无符号型	register	寄存器	switch	开关
signed	有符号型	typedef	类型定义	default	默认
struct	结构式	if	如果	case	情况
union	共用式	else	否则	sizeof	计算字节数
enum	枚举式	for	表示一种循环		

3. 标识符及其命名规则 在 C 语言中, 用户为了区分程序中出现的常量、变量、函数和数组等, 就要给它们取不同的名字。组成名字的字符序列, 称为标识符。即标识符是用来表示程序中的变量名、函数名、数组名等用途的字符序列。在 C 语言中, 标识符的命名必须遵循以下规则:

- 1) 标识符只能由字母 A~Z、a~z、数字 0~9 或下画线 _ 组成。
- 2) 标识符的第一个字符不能是数字。
- 3) 关键字不能作为标识符使用。

【例 1.4】 试判断下面所给出的字符序列, 哪些是正确的 C 语言标识符。

x _906 A203 B_1 C.508 int _1a
8b? y-56 b_1 a2.abc AB.txt 1.doc 1a_

解: 根据构成标识符的命名规则可知, 上述字符序列里, 正确的标识符为

x _906 A203 B_1 _1a b_1

注意: 由于 C 语言程序区分大小写, 所以 B_1 和 b_1 是两个不同的标识符。

1.4 C 语言程序的开发过程及上机步骤

C 语言是一种编译型的程序设计语言。用 C 语言开发程序, 需要一个开发环境。目前流行的 C 语言有: Microsoft C 或称 MS C、Borland Turbo C 或称 Turbo C、AT&T C。现在主要以 Turbo C 2.0 为 C 语言开发环境, 来介绍 C 语言程序的上机操作过程。

1.4.1 C 语言程序的开发过程

开发一个 C 语言程序要经过编辑、编译、连接和运行 4 个步骤, 才能得到运行结果。

1. 编辑 编辑是指将 C 语言源程序通过键盘输入到计算机中去, 并加以修改, 最后以文本文件的形式存入磁盘。文件名由用户自行定义, 扩展名为 .c, 把这个文件称为源程序文件。

2. 编译 编译是把 C 语言程序翻译成二进制的目标程序。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, 当发现这类错误时, 就将错误的类型和位置显示出来, 以帮助用户修改错误。如果未发现错误, 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, 扩展名为 .obj。

3. 连接 连接也称为链接或装配, 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。可执行的目标程序的扩展名 .exe 由系统自动确定。

4. 运行 直接运行可执行的目标程序, 即可得到程序的运行结果。通常, 在 DOS 环境下直接输入可执行文件名, 在 Windows 环境下双击可执行文件名即可运行程序。

上述 4 步骤如图 1-1 所示。图中, 实线箭头表示操作流程, 虚线箭头表示操作所需的条件和产生的结果。

1.4.2 Turbo C 2.0 集成环境介绍

Turbo C 2.0 是美国 Borland 公司开发的一个 C 语言集成环境。它集程序编辑、编译、连接、调试和运行为一体, 具有速度快、效率高和功能强等优点, 使用非常方便。

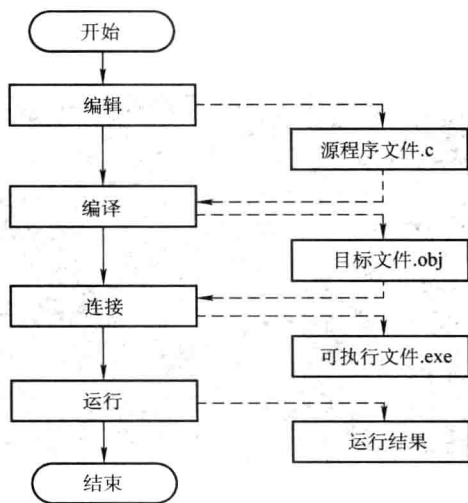


图 1-1 C 语言程序的开发

为了使用 Turbo C 2.0，必须先将其安装在计算机硬盘的某一目录下。

1. 启动 Turbo C 2.0 Turbo C 2.0 可以在 DOS 环境下启动，也可以在 Windows 环境下启动。

(1) 在 DOS 环境下的启动：启动计算机，切换到 DOS 环境下，如果需要使用汉字，则可启动汉字系统（一般用 UC DOS），然后进入到 TC 目录中，再执行 TC. EXE 程序。假定 Turbo C 2.0 安装在 C 盘根目录下，则可用命令：

```
C > CD C:\TC
```

```
C > TC
```

启动 TC 后，即可进入到 Turbo C 2.0 集成环境。

(2) 在 Windows 环境下的启动：在 Windows 环境下，打开“我的电脑”，找到 Turbo C 所在的文件夹“TC”，打开该文件夹，用鼠标双击“TC”文件夹中的“TC. EXE”文件。

2. Turbo C 2.0 集成环境窗口 启动 Turbo C 2.0 之后，Turbo C 2.0 的主窗口将呈现出来，如图 1-2 所示。

主窗口的中间位置给出了 Turbo C 2.0 的版本信息，按任意键，版本信息即可消失。主窗口由主菜单栏、编辑区、信息区和功能键提示行 4 部分组成。

(1) 主菜单：提供了 Turbo C 2.0 集成环境的所有功能，共 8 个菜单项。

File（文件）菜单：用于建立、装入和存储程序文件，处理目录、切换 DOS 环境、以及退出 Turbo C 2.0 环境。

Edit（编辑）菜单：用于建立和编辑、修改源程序。

Run（运行）菜单：用于运行当前内存中的源程序和查看运行结果。

Compile（编译）菜单：用于编译、连接当前内存中的源程序。

Project（项目）菜单：用于支持大型多程序文件的开发和维护。

Options（可选项）菜单：用于设置 Turbo C 2.0 集成环境的各种操作。

Debug（调试）菜单：用于设置调试选项，进行调试操作。

Break/watch（断点/监视）菜单：用于对正在执行的程序指定表达式进行监视以及设置