



21st CENTURY

实用规划教材

21世纪全国高职高专
计算机系列实用规划教材

C语言 程序设计(第2版)

主 编 刘迎春 王 磊
副主编 陈 静 陈庆惠 马玉凤



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国高职高专计算机系列实用规划教材

C 语言程序设计 (第 2 版)

主 编 刘迎春 王 磊
副主编 陈 静 陈庆惠 马玉凤



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书是 2005 年山东省精品课程建设的成果。全书共分为 11 章, 内容包括: C 语言概述, 数据类型、运算符与表达式, 顺序结构程序设计, 选择结构程序设计, 循环结构程序设计, 数组, 函数, 指针, 结构体, 位运算和文件。本书对带宏、条件编译等很少使用的内容不作讲解, 重点强化选择结构、循环结构、数组以及函数等编程中经常使用的知识点。

全书采用任务驱动式教学, 创设任务情景, 先提出任务, 带着任务学习相关知识点, 然后解决任务, 最后进行任务的深化。在教学内容的组织上, 基本语句、基本语法够用即可, 重视算法思想的讲解, 侧重培养学生的逻辑思维能力和编程解决实际问题的能力。

本书既可作为高等院校计算机及相关专业的教材, 也适合作为自学教材以及 C 程序开发人员的参考书, 还可以作为全国计算机等级考试的培训教材。

图书在版编目(CIP)数据

C 语言程序设计/刘迎春, 王磊主编. —2 版. —北京: 北京大学出版社, 2009.8

(21 世纪全国高职高专计算机系列实用规划教材)

ISBN 978-7-301-15476-2

I. C… II. ①刘…②王… 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(2009)第 116497 号

书 名: C 语言程序设计(第 2 版)

著作责任者: 刘迎春 王 磊 主编

策 划 编 辑: 李彦红

责 任 编 辑: 魏红梅

标 准 书 号: ISBN 978-7-301-15476-2/TP · 1034

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电子邮箱: pup_6@163.com

印 刷 者: 河北滦县鑫华书刊印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 20.25 印张 472 千字

2006 年 8 月第 1 版 2009 年 8 月第 2 版 2009 年 8 月第 1 次印刷

定 价: 32.00 元

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

第2版前言

第1版《C语言程序设计》教材自2006年8月出版,已经两年半了,有3届学生相继使用,在北京大学出版社“2008年高职高专规划教材”评选中,获一等奖。为了更好地适应现在教学的需要,编者重新修订了该教材,对部分不重要的内容进行了调整,对引领知识点的任务、情景进行了修改,使其更具有真实工作任务和情景的效果,但原书的基本宗旨和风格没变,采用任务驱动式教学,创设任务情景,将所有必讲内容分为30讲。每讲先提出任务,带着任务学习相关知识点,然后解决任务,最后进行任务的深化。

全书分为11章,第1、2章,主要是C语言基础部分,包括C语言程序的组成结构、C语言数据类型、运算符和表达式以及输入和输出等;第3~5章,主要是C语言的顺序、选择和循环结构;第6~8章,是C语言最重要的编程技术,即数组、函数和指针,也是学生在学习中感到困难的部分;第9章是结构体;第10章是位运算;第11章是文件。

在修订中,以培养学生编程能力为目标,确定知识点的选择,重点突出循环、数组、函数的使用和编程训练,对不经常使用的内容删节、合并,如:删除了2.6节变量赋初值;3.1节C语句概述;3.2节赋值语句等,将涉及到的个别知识,合并到相关章节。增加了循环编程举例,强化学生的编程能力的培养。

本书学时安排为90学时,也可根据学生情况具体调整。

全书修订中,第1~3章,由刘迎春老师负责修订;第4、9章,由马玉凤老师负责修订;第5、6章,由王磊老师负责修订;第8、11章,由陈静老师负责修订;第7、10章由陈庆惠老师负责修订。全书由刘迎春、王磊老师审核,刘迎春老师最终统稿。

我们在修订中,尽可能征求任课教师意见,力求更适合学生和教师使用,但由于水平有限,不足之处,希望各位老师和同学指正,不胜感激。

主编的联系方式: jnycliu@163.com 或 sdjnw@126.com。

衷心感谢所有关心本书编写的老师和朋友。

本书提供以下教学资源:

(1) 配套课件(<http://www.pup6.com>)。

(2) Flash 算法动画。

(3) 源代码。

(4) 课程设计选题及设计。

(5) 考试试卷23套。

(6) C语言精品课程论坛(<http://www.jnzjxy.com.cn>)。

(7) 在线测试(<http://www.jnzjxy.com.cn>)。

编 者

2009年3月

目 录

第1章 C语言概述.....1	2.6.2 自动转换.....39
1.1 C语言出现的历史背景.....1	2.7 算术运算符和算术表达式.....41
1.2 C语言的特点.....2	2.7.1 C运算符简介.....41
1.3 简单的C语言程序介绍.....4	2.7.2 算术运算符和算术表达式.....42
1.3.1 三个简单的C语言程序实例.....4	2.8 赋值运算符和赋值表达式.....47
1.3.2 C语言程序的基本组成.....6	2.8.1 赋值运算符与赋值表达式.....47
1.4 C语言程序的上机步骤.....8	2.8.2 复合的赋值运算符.....50
1.5 算法.....9	2.9 逗号运算符和逗号表达式.....51
1.5.1 算法的概念.....9	2.10 本章小结.....53
1.5.2 简单算法举例.....10	第3章 顺序结构程序设计.....57
1.5.3 算法的特性.....12	3.1 输入/输出的概念及其C语言的实现.....57
1.5.4 算法的表示.....13	3.2 字符数据的输入/输出.....58
1.5.5 结构化程序设计方法.....18	3.2.1 putchar函数——字符输出函数.....58
1.6 本章小结.....21	3.2.2 getchar函数——字符输入函数.....58
第2章 数据类型、运算符与表达式.....22	3.2.3 putch函数——字符输出函数.....59
2.1 C语言的数据类型.....22	3.2.4 getch函数——字符输入函数.....59
2.2 常量与变量.....24	3.3 格式输入与输出.....61
2.2.1 常量与符号常量.....24	3.3.1 printf函数——格式输出函数.....61
2.2.2 变量.....24	3.3.2 scanf()函数——格式输入函数.....67
2.3 整型数据.....27	3.4 顺序结构程序设计举例.....71
2.3.1 整型常量的表示方法.....27	3.5 预处理命令.....75
2.3.2 整型变量.....27	3.5.1 宏定义.....75
2.4 实型数据.....31	3.5.2 文件包含.....78
2.4.1 实型常量的表示方法.....31	3.6 本章小结.....80
2.4.2 实型变量.....32	第4章 选择结构程序设计.....82
2.5 字符型数据.....34	4.1 关系运算符和关系表达式.....82
2.5.1 字符常量.....34	4.1.1 关系运算符及其优先级.....82
2.5.2 字符变量.....36	
2.5.3 字符数据在内存中的存储形式及其使用.....36	
2.5.4 字符串常量.....38	
2.6 各类数值型数据之间的混合运算.....39	
2.6.1 整型、实型、字符型数据之间可以混合运算.....39	

4.1.2 关系表达式	83	6.2.3 二维数组的初始化	153
4.2 逻辑运算符和逻辑表达式	84	6.3 字符数组	157
4.2.1 逻辑运算符及其优先级	84	6.3.1 字符数组的定义	157
4.2.2 逻辑表达式	86	6.3.2 字符数组的初始化	158
4.3 单分支和双分支选择语句	87	6.3.3 字符数组的引用	158
4.3.1 单分支选择语句	87	6.3.4 字符串和字符串结束标志	159
4.3.2 双分支选择语句	88	6.3.5 字符数组的输入/输出	160
4.4 多分支选择语句	93	6.4 字符串处理函数	166
4.4.1 if...else...多分支选择语句	93	6.5 本章小结	174
4.4.2 switch 开关语句	95	第 7 章 函数	177
4.5 选择语句的嵌套与条件运算符	97	7.1 函数的定义、函数参数和函数值	177
4.5.1 选择语句的嵌套	97	7.1.1 C 语言对函数的规定	177
4.5.2 条件运算符	98	7.1.2 函数的定义	178
4.6 本章小结	103	7.2 函数的调用	180
第 5 章 循环结构程序设计	106	7.3 函数的嵌套调用	185
5.1 概述	106	7.4 函数的递归调用	187
5.1.1 基本概述	106	7.5 数组作为函数参数	191
5.1.2 goto 语句	107	7.5.1 变量存取的实质	191
5.2 while 语句	107	7.5.2 函数调用的两种方式	192
5.3 do...while 语句	109	7.5.3 数组元素作为函数实参	192
5.4 for 语句	116	7.5.4 数组名作为函数实参	193
5.5 几种循环的比较	121	7.6 局部变量和全局变量	199
5.5.1 循环结构的基本组成部分	121	7.6.1 局部变量	199
5.5.2 几种循环的比较	121	7.6.2 全局变量	201
5.6 循环的嵌套	122	7.7 动态存储变量和静态存储变量	202
5.7 break 语句和 continue 语句	128	7.7.1 变量的存储类别	202
5.7.1 break 语句	128	7.7.2 局部变量的存储方式	203
5.7.2 continue 语句	129	7.7.3 全局变量的存储方式	205
5.8 循环结构程序设计举例	134	7.8 内部函数和外部函数	205
5.9 本章小结	140	7.8.1 内部函数	205
第 6 章 数组	143	7.8.2 外部函数	206
6.1 一维数组的定义和引用	143	7.9 本章小结	210
6.1.1 一维数组的定义	143	第 8 章 指针	213
6.1.2 一维数组元素的引用	144	8.1 指针与指针变量	213
6.1.3 一维数组的初始化	145	8.1.1 指针的实质是地址	213
6.2 二维数组的定义和引用	151	8.1.2 指针变量	214
6.2.1 二维数组的定义	151	8.1.3 指向变量的指针变量	216
6.2.2 二维数组的引用	152	8.1.4 指针变量的应用	218

8.2 一维数组与指针变量	222	9.8 枚举类型	271
8.2.1 指向一维数组和数组元素 的指针变量	222	9.9 用 typedef 定义类型	273
8.2.2 指针变量作为函数参数	225	9.10 本章小结	274
8.2.3 数组名作为函数参数	227	第 10 章 位运算	277
8.3 二维数组与指针变量	232	10.1 位运算符与位运算	277
8.3.1 二维数组名的含义	232	10.1.1 按位与(&)	278
8.3.2 指向二维数组名的指针 变量	234	10.1.2 按位或()	279
8.4 字符串与指针变量	237	10.1.3 按位异或(^)	279
8.4.1 字符串的表示形式	237	10.1.4 按位取反(~)	280
8.4.2 指向字符串的指针作为函数 参数	240	10.1.5 左移位(<<)	280
8.5 指针变量的其他应用形式	245	10.1.6 右移位(>>)	281
8.5.1 指针函数	245	10.2 本章小结	285
8.5.2 指针数组	246	第 11 章 文件	287
8.5.3 指向指针的指针	246	11.1 C 语言文件概述	287
8.6 本章小结	250	11.2 文件的打开与关闭	288
第 9 章 结构体	253	11.2.1 FILE 类型	288
9.1 定义结构体类型	254	11.2.2 文件的打开	289
9.2 定义结构体类型变量的方法	255	11.2.3 文件的关闭	290
9.3 结构体变量的引用	256	11.3 文件的读/写	291
9.4 结构体变量的初始化	257	11.3.1 fputc 函数和 fgetc 函数	291
9.5 结构体数组	258	11.3.2 fread 函数和 fwrite 函数	294
9.6 指向结构体类型数据的指针	258	11.4 其他的文件读/写函数	300
9.6.1 结构体指针变量	258	11.5 文件的定位	300
9.6.2 结构体变量作为函数参数	259	11.5.1 feof 函数	300
9.7 用指针处理链表	263	11.5.2 rewind 函数	301
9.7.1 链表概述	263	11.5.3 fseek 函数	301
9.7.2 简单链表的建立	264	11.6 本章小结	303
9.7.3 处理动态链表所需的函数	265	附录 A 常用字符与 ASCII 代码 对照表	305
9.7.4 建立动态链表	266	附录 B 关键字及其用途	306
9.7.5 输出链表	268	附录 C 运算符的优先级和结合性	307
9.7.6 链表的插入操作	268	附录 D Turbo C 2.0 常用库函数	309
9.7.7 链表的删除操作	269	参考文献	314
9.7.8 链表的综合操作	270		

第 1 章 C 语言概述

C 语言是国际上广泛流行的计算机高级程序设计语言，它集高级语言和低级语言的功能于一体，除了可用于操作系统的开发，也适合于工业控制、智能仪表、嵌入式系统和硬件驱动等方向。同时它还具有效率高和可移植性强等特点，因此被称为当代最优秀的程序设计语言。本章详细介绍 C 语言的发展和特点、C 语言程序的结构、C 语言程序的上机执行过程以及程序的灵魂——算法。

本章内容

- (1) C 语言出现的历史背景。
- (2) C 语言的特点。
- (3) C 语言程序的结构。
- (4) C 语言程序的上机执行过程。
- (5) 算法的概念及特点。
- (6) 算法的表示方法。
- (7) 结构化程序设计方法。

相关知识点

1.1 C 语言出现的历史背景

1. C 语言的发展历史

C 语言的产生，起源于对系统程序设计的深入研究和探索。1967 年，英国剑桥大学的 M. Richards 在 CPL(Combined Programming Language)语言的基础上，实现并推出了 BCPL(Basic Combined Programming Language)语言。

1970 年，美国贝尔实验室的 K.Thompson 以 BCPL 语言为基础，设计了一种类似于 BCPL 的语言，称为 B 语言。他用 B 语言在 PDP-7 机上实现了第一个实验性的 UNIX 操作系统。

1972 年，贝尔实验室的 Dennis M. Ritchie 为克服 B 语言的诸多不足，在 B 语言的基础上重新设计了一种语言，由于是 B 语言的后继，故称为 C 语言。1973 年，贝尔实验室的 K. Thompson 和 Dennis M. Ritchie 合作，首先用 C 语言重新改写了 UNIX 操作系统，在当时的 PDP-11 计算机上运行。此后，C 语言作为 UNIX 操作系统上标准的系统开发语言，伴随着 UNIX 操作系统的发展，越来越广泛地被人们接受和应用并被移植到其他计算机系统。

1978 年, Brian W. Kernighan 和 Dennis M. Ritchie(K&R)正式出版了著名的 *The C Programming Language* 一书, 此书中介绍的 C 语言成为后来广泛使用的 C 语言版本基础, 它被称为标准 C 语言。

C 语言的标准化工作是从 20 世纪 80 年代初期开始的。1983 年, 美国国家标准化协会 (ANSI, American National Standard Institute) 根据各种 C 语言版本对 C 语言扩充和发展, 颁布了 C 语言的新标准 ANSI C。ANSI C 比标准 C 有了很大的扩充和发展。

由于 C 语言的不断发展, 1987 年, 美国国家标准化协会在综合各种 C 语言版本的基础上, 又颁布新标准, 为了与标准 ANSI C 区别, 称为 87 ANSI C。1990 年, 国际标准化组织 ISO 接受了 87 ANSI C 作为 ISO C 的标准。这是目前功能最完善、性能最优良的 C 语言新版本。目前流行的 C 语言编译系统都是以它为基础的。本书讲述的内容基本上是以 ANSI C 为基础, 并参考 87 ANSI C。

2. 目前我国 PC 系列兼容机上常用的 C 语言版本

(1) Borland 公司: Turbo C(V2.0, V3.0)、Turbo C++、Borland C++和 C++Builder(Windows 版本)。

(2) Microsoft 公司: Microsoft C(V5.0, V6.0, V7.0)和 Visual C++(Windows 版本)。

(3) Computer Innovations 公司: C 86(V2.3)。

(4) Lattice 公司: Lattice C(V4.0)等。

1.2 C 语言的特点

1. C 语言的特点

C 语言之所以能全球广泛流行, 正是由于自身具备的突出特点, 从语言体系和结构上看, 是结构化程序设计语言; 从计算机语言分类上看, 既有高级语言的特点, 又有低级语言的功能。

(1) C 语言既有高级语言特点, 又有低级语言的功能。

计算机语言的发展经历了机器语言、汇编语言和高级语言三个阶段。

机器语言是最底层的计算机语言, 只包含 0、1 组成的二进制代码构成的操作指令。机器语言编写的指令系统, 可以被计算机系统直接执行, 且效率高。但机器语言难学、难用, 不容易移植。

汇编语言是用助记符代替机器语言中的操作码, 用十进制或十六进制数表示操作数, 对机器语言进行简单的符号化。与机器语言相比, 易于记忆、理解, 但仍是面向机器的低级语言, 通用性差。

高级语言是一种接近于自然语言和数学语言的计算机语言, 编程时不需要考虑机器的内部结构和存储单元的分配。一个语句相当于许多条计算机指令, 描述方式也与日常处理问题的逻辑思维习惯相接近, 因此易于学习和掌握。高级语言具有良好的通用性和可移植性。C、Basic 和 Pascal 都是高级语言。

C 语言是面向问题的程序设计语言，独立于计算机的硬件，对具体的算法进行描述，它的特点是独立性，通用性和可移植性好；而且 C 语言有高级语言的结构和语句，因此，C 语言具有高级语言的特点。

C 语言还可以像低级语言汇编一样对计算机存储介质中的位、字节和地址进行操作，实现对计算机硬件的控制，尤其是 C 语言中可以嵌入汇编子程序，使 C 语言具有强大的底层控制能力，成为 21 世纪最受欢迎的编程语言。

(2) C 语言是结构化程序设计语言。

结构化程序语言的特点是，代码和数据的分割化，即程序的各个部分除了必要的信息交流外彼此独立，这种结构化方式可使程序层次清晰，便于使用、维护和调试。C 语言是函数式的语言，编写程序就是编写函数，各个函数除了必要的数据交流(数据、地址传递和返回)，功能相对独立。另外，C 系统还提供了系统库函数，供用户调用。C 语言有完整的选择、循环结构语句，从而使得程序完全结构化。

(3) C 语言数据结构丰富，功能强大。

C 语言数据类型丰富，支持几乎所有数据的存储、计算和输出；并引入了指针的概念，可使程序效率更高。而且，C 语言计算功能、逻辑判断功能也比较强大，可以实现决策目的。另外，C 语言具有强大的图形功能，支持多种显示器和驱动器。

(4) C 语言应用范围广，适应性强。

C 语言还有一个突出的优点就是适合多种操作系统，如 DOS、UNIX，也适用于多种机型，能适应从 8 位微型机到巨型机的所有机种。

除此之外，C 语言还具有简洁、灵活，便于学习和应用，程序运行质量、效率高等特点。

2. C 语言是当代最优秀的程序设计语言

有人称 C 语言为“王者”，有人称 C 语言是程序员使用的语言，更有甚者，称 C 语言为“母语”。

自 1978 年贝尔实验室正式发布 C 语言以来，C 语言以简洁紧凑的风格，面向过程的编程方式，丰富的数据结构和强大的底层控制能力获得迅速发展。到 20 世纪 80 年代，C 语言已经成为最受欢迎的编程语言，许多著名的系统软件，如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的；用 C 语言加上一些汇编语言子程序，就更能显示 C 语言的威力，像 PC-DOS、Linux 等；到 20 世纪 90 年代，C 语言已经成为计算机专业学生的首选教学语言，并成为一代程序员的主要编程语言。

进入 21 世纪，随着个人电子消费产品和开源软件(源代码开放的软件，如 Linux)的流行，C 语言再次焕发生机。由于 C 语言在底层控制和性能方面的优势，使之成为芯片级开发(嵌入式)和 Linux 平台开发的首选语言；在通信、网络协议、破解、3D 引擎、操作系统、驱动、单片机、手机、PDA(Personal Digital Assistant，即个人数码助理，如掌上电脑)、多媒体处理、实时控制等领域，C 语言正在用一行行代码证明它从应用级开发到系统级开发的强大和高效。

据 TIOBE(The Importance Of Being Earnest 编程语言排行榜)数据显示, 2008 年, C 语言稳居第二的位置, 有望成为 2008 年年度语言(至截稿时, TIOBE 尚未公布)。

注: TIOBE 公司成立于 2000 年 10 月 1 日, 由瑞士公司 Synspace 和一些独立的投资人创建。该公司主要从事软件质量的评估和跟踪。TIOBE 公司推出了 TIOBE 编程语言排行榜, 将程序设计语言以排名列表的形式提供出来, 并且每个月更新一次, 用来表示程序设计语言的流行度。2008 年 12 月发布的编程语言前三名是: Java, C, C++。该排行榜是通过统计该编程语言在主流搜索引擎(如百度、Google、雅虎等)上被搜索的次数来计算的。

1.3 简单的 C 语言程序介绍

1.3.1 三个简单的 C 语言程序实例

用 C 语言语句编写的程序称为 C 语言程序或 C 语言源程序。本节从三个简单的 C 语言程序中分析 C 语言程序的基本组成。

【例 1.1】 在窗体上显示 “Welcome to the C world!” 语句, 并且显示一个由 “*” 号组成的大写的 C 字母。

实现此功能的 C 语言程序参考源代码如下:

```
#include <stdio.h>                /*文件包含*/
main()                            /*定义主函数*/
{
    printf("Welcome to the C world!\n"); /*输出欢迎语句*/
    printf("*****\n");                /*以下几行输出大写字母 C*/
    printf("**\n");
    printf("**\n");
    printf("*****\n");
}
```

说明: (1) main 称为 “主函数”。每个 C 语言程序都必须有一个 main 函数, 它是每一个 C 语言程序的执行起始点(入口点)。

(2) 用 { } 括起来的是 “主函数” main 的函数体。main 函数中的所有操作(语句)都在这一对 { } 之间。即 main 函数的所有操作都在 main 函数体中。

(3) printf 是 C 语言的输出函数, 功能是用于程序的输出(显示在屏幕上), 双引号内的字符串原样输出。“\n” 是换行符, 即在输出完 “Welcome to the C world!” 或 “*” 后回车换行。

(4) 每条语句用 “;” 号结束。

(5) /*……*/ 括起来的部分是一段注释, 注释只是为了改善程序的可读性, 在编译、运行时不起作用(事实上编译时会跳过注释, 目标代码中不会包含注释)。注释可以放在程序任何位置, 并允许占用多行, 只是需要注意 “/*” 和 “*/” 匹配。

思考:

根据此例编写程序, 在屏幕上输出如下内容。

```
*****
How are you!
*****
```

【例 1.2】 计算两数之乘积，并输出结果。

实现此功能的 C 语言程序参考源代码如下：

```
main()          /*计算两数之积*/
{
    int a,b,p;    /*这是定义变量*/
    a=12;b=25;    /*以下 3 行为 C 语句*/
    p=a*b;
    printf("p=%d\n",p);
}
```

说明：(1) 本程序包含一个 main 函数作为程序执行的起点；{ } 内为 main 函数的函数体，main 函数的所有操作均在 main 函数体中。

(2) “int a,b,p;” 是变量声明语句；声明了三个具有整数类型的变量 a,b,p；C 语言的变量必须先声明再使用。

(3) “a=12; b=25;” 是两条赋值语句。将整数 12 赋给整型变量 a，将整数 25 赋给整型变量 b。注意这是两条赋值语句，每条语句均用“;”结束。
也可以将两条语句写成两行，即

```
a=12;
b=25;
```

(4) “p=a*b;” 是将 a,b 两变量内容相乘，然后将结果赋值给整型变量 p。

(5) “printf(“p=%d\n”, p);” 是调用库函数 printf 输出 p 的结果。“%d” 为格式控制字符串，表示 p 的值以十进制整数形式输出。

程序运行后，输出：

```
p=300
```

思考：

根据此例编写程序，计算小明本学期四门功课的平均分。

【例 1.3】 输入两个整数，找出二者中较大的数，并输出。

实现此功能的 C 语言程序参考源代码如下：

```
/*用户自定义函数 max()，实现从两个数中找大数 */
int max(int x,int y)      /*计算两数中较大的数*/
{                          /*max 函数体开始*/
    int z;                /*声明部分，定义变量*/
    if(x>y) z=x;
    else z=y;
    return z;             /*将 z 值返回，通过 max 带回调用处*/
}                          /*max 函数体结束*/
```

```
/* 主函数*/
main()
{
    int a,b,c;
    scanf("%d,%d",&a,&b);
    c=max(a,b);
    printf("max=%d\n",c);
}

/*main 函数体开始*/
/*声明部分定义变量*/
/*调用 max, 将调用结果赋给 c*/
/*输出 c 的值*/
/*main 函数体结束*/
```

说明: (1) 本程序包括两个函数。其中主函数 main 仍然是整个程序执行的起点。函数 max 是用户自己定义的函数, 该函数功能是找出两数中较大的数。

(2) scanf 是 C 语言的输入函数。它的作用是从键盘上输入 a, b 的值。“%d”为格式控制字符串, 表示 a, b 的值以十进制整数形式输入, &a 和 &b 中的“&”的含义是“取地址”, 此 scanf 函数的作用是将键盘输入的两个数值分别输入到变量 a 和 b 的地址所标志的单元中, 即输入给变量 a, b。

(3) 主函数 main 调用用户自定义函数 max 获得两个数字中较大的值, 并赋给变量 c。最后输出变量 c 的值(结果)。

(4) 函数 max 同样也用 {} 将函数体括起来。

(5) max 函数的定义和功能, 在第 7 章函数还会详细讲解, 读者只要了解: 一个 C 语言程序除了 main 函数外, 还可以有用户自定义的函数。

1.3.2 C 语言程序的基本组成

1. C 语言程序由函数构成

(1) 一个 C 语言程序至少包含一个 main() 函数, 也可以包含一个 main() 函数和若干个其他函数。函数是 C 语言程序的基本单位。

(2) 被调用的函数可以是系统提供的库函数, 如上面例子中涉及的输出函数 printf 和输入函数 scanf; 也可以是用户根据需要自己编写设计的函数, 如 max。C 程序设计语言是函数式的语言, 程序的全部工作都是由各个函数完成。编写 C 语言程序就是编写一个个函数。

2. main 函数是每个程序执行的起始点

一个 C 语言程序总是从 main 函数开始执行, 而不管 main() 函数在程序中的位置。可以将 main 函数放在整个程序的最前面, 也可以放在整个程序的最后, 或者放在其他函数之间。

3. 一个函数由函数首部和函数体两部分组成

(1) 函数首部是一个函数的第一行。

例如: 例 1.3 中的

```
int max (int x, int y)
```

就是函数首部。

(2) 函数体是函数首部下用一对 {} 括起来的部分。

例如：例 1.3 中的

```
/*函数 max 的函数体*/
{
    int z;
    if(x>y) z=x;
    else z=y;
    return z;
}
```

如果函数体内有多个 {}, 最外层是函数体的范围。函数体一般包括声明部分、执行部分两部分。

声明部分：在这部分定义本函数所使用的变量。

执行部分：由若干条语句组成的命令序列(可以在其中调用其他函数)。

4. C 语言程序书写格式自由

- (1) 一行可以写几个语句，一个语句也可以写在多行上。
- (2) 程序没有行号。
- (3) 每条语句的最后必须有一个分号“;”表示语句的结束。

5. 可以使用 /*……*/ 对 C 语言程序中的任何部分作注释

注释可以提高程序可读性，使用注释是编程人员的良好习惯。

(1) 实践中，写好的程序往往需要修改、完善，事实上没有一个应用系统是不需要修改、完善的。很多人会发现自己编写的程序在经历了一些时间以后，由于缺乏必要的文档、必要的注释，最后连自己都很难再读懂。需要花费大量时间重新思考、理解原来的程序。如果一开始编程就对程序进行注释，虽然刚开始麻烦一些，但日后可以节省大量的时间。

(2) 一个实际的系统往往是多人合作开发，程序文档、注释是其中重要的交流工具。

6. C 语言本身不提供输入/输出语句

输入/输出的操作是通过调用库函数(scanf, printf 等)完成的。

7. C 程序的一般组成形式

```
main()                /*主函数首部*/
{ 变量定义            /*主函数体*/
  执行语句组
}
子函数名 1(参数)      /*子函数首部*/
{ 变量定义            /*子函数体*/
  执行语句组
}
子函数名 2(参数)      /*子函数首部*/
{ 变量定义            /*子函数体*/
  执行语句组
}
.....
```

```

子函数名 N(参数)      /*子函数首部*/
{ 变量定义            /*子函数体*/
  执行语句组
}

```

其中, 子函数名 1 至子函数名 N 是用户自定义的函数。

1.4 C 语言程序的上机步骤

编写出 C 语言程序仅仅是程序设计工作中的第一步, 写出来的 C 语言程序需要在 C 语言的编译系统中调试运行, 才能得到正确的运行结果。

目前在微机上常用的 C 语言编译系统中, Borland International 公司的 Turbo C 2.0 和 Microsoft 公司的 Microsoft VC++6.0、Quick C 等都被广泛使用。

本书在配套的《C 语言程序设计上机指导与同步训练》中, 对 Turbo C 2.0 和 Microsoft VC++6.0 编译环境给出介绍和使用指导。

本书所有的实例, 未加特殊声明, 均在 Turbo C 2.0 中编译通过。

C 语言程序的上机执行过程一般要经过编辑、编译、连接、运行 4 个步骤, 如图 1.1 所示。

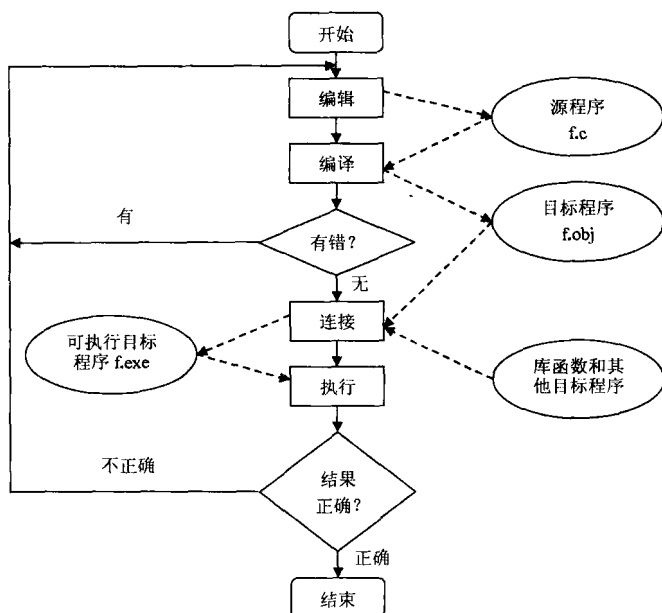


图 1.1 C 语言程序的上机步骤

1. 编辑 C 语言源程序

编辑是用户把编写好的 C 语言源程序输入计算机, 并以文件的形式存放在磁盘上。其标识为“文件名.c”。其中文件名是由用户指定的符合 C 语言标识符规定的任意字符组成, 扩展名要求为“.c”, 表示是 C 语言源程序。例如 file1.c、t.c 等。

2. 编译C语言源程序

编译是把C语言源程序翻译成用二进制指令表示的目标文件。编译过程由C语言编译系统提供的编译程序完成。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,就将错误的类型和所在的位置显示出来提供给用户,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。目标程序的文件标识为“文件名.obj”。

3. 程序连接

目标程序计算机还是不能执行的。程序连接过程是用系统提供的连接程序(也称链接程序或装配程序)将目标程序、库函数或其他目标程序连接装配成可执行的程序。可执行程序的文件名为“文件名.exe”。

4. 运行程序

运行程序是指将可执行的目标程序投入运行,以获取程序处理的结果。如果程序运行结果不正确,可重新回到第一步,对程序进行编辑修改、编译和运行。与编译、连接不同的是,运行程序可以脱离语言处理环境。因为它是对一个可执行程序进行操作,与C语言本身已经没有联系,所以可以在语言开发环境下运行,也可直接在操作系统下运行。

注意:对不同型号计算机上的C语言版本,上机环境各不相同,编译系统支持性能各异,

上述步骤有些可再分解,有些也可集成进行批处理,但逻辑上是基本相同的。

提出问题

问题 1. 如果想将 A 瓶中的红墨水和 B 瓶中的蓝墨水对调一下,即 A 瓶中装蓝墨水, B 瓶中装红墨水,如何实现?

问题 2. 如果用计算机来计算 $1+2+3+4+5+\cdots+100$ 的值,应怎样设计?

问题 3. 我国数学家秦九韶在《算书九章》一书中曾记载了求两个正整数 m 和 $n(m \geq n)$ 最大公约数的方法,即采用辗转相除法(也称欧几里德算法)。计算机中这种思想如何实现?

相关知识点

1.5 算 法

1.5.1 算法的概念

读者都听说过数学家高斯小时候的一则故事:老师让大家计算 $1+2+3+4+\cdots+100$ 的值,

当大部分小朋友还在纸上 $1+2$, 然后再加 3, 再加 4, …… 计算时, 高斯已经得出了答案 5050, 他采用了 $(1+100)+(2+99)+\cdots+(49+52)+(50+51)=101\times 50=5050$ 方法, 快速得出了正确答案, 获得了老师的表扬。对于这个问题, 高斯和小朋友们采用了不同的解决问题的方法, 这种解决问题的方法, 在程序设计中被称为算法。

一个程序应包括以下两方面的内容。

(1) 对数据的描述。在程序中要指定数据的类型和数据的组织形式, 即数据结构。

(2) 对操作的描述。即操作步骤, 也就是算法。

著名计算机科学家沃思提出一个公式: 数据结构+算法=程序。

实际上, 一个程序除以上两个主要要素之外, 还应采用结构化程序设计方法进行程序设计, 并且用某一种计算机语言表示, 因此, 可以这样表示:

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境

也就是说, 以上 4 个方面是一个程序设计人员所应具备的知识。在设计一个应用程序时要综合运用这几个方面的知识。在这 4 个方面中算法是灵魂, 数据结构是加工对象, 语言是工具, 编程需要采用合适的方法。

算法——就是为了解决一个问题而采取的方法和步骤, 是程序的灵魂。程序设计就是设计合理的数据结构和设计对数据结构进行操作的算法, 回答“做什么”和“怎么做”的两个问题。

解决同一问题可能有多种算法, “好”的算法可以提高处理问题的效率, 减少对系统资源的占用。相反, “不好”的算法会降低处理的效率, 占用大量的系统资源, 而错误的算法则会导致错误的处理结果。

不是只有计算问题才有算法。例如, 加工一张写字台, 其加工顺序是: 桌腿→桌面→抽屉→组装, 这就是加工这张写字台的算法。当然, 如果是按“抽屉→桌面→桌腿→组装”这样的顺序加工, 那就是加工这张写字台有另一种算法, 这其中没有计算问题。

计算机算法分为两大类: 数值运算算法和非数值运算算法。数值运算的目的是求数值解, 例如求方程的根和求一个函数的定积分等, 都属于数值运算范围。非数值运算包括的面十分广泛, 最常见的是用于事务管理领域, 例如图书检索、人事管理等。

解决问题

1.5.2 简单算法举例

下面通过三个简单的问题设计算法的思维方法。

【例 1.4】(问题 1) 如果想将 A 瓶中的红墨水和 B 瓶中的蓝墨水对调一下, 即 A 瓶中装蓝墨水, B 瓶中装红墨水, 如何实现?

算法思想: 这是一个非数值运算问题。因为两个墨水瓶中的墨水不能直接交换, 所以解决这一问题的关键是引入第三个瓶子。