

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

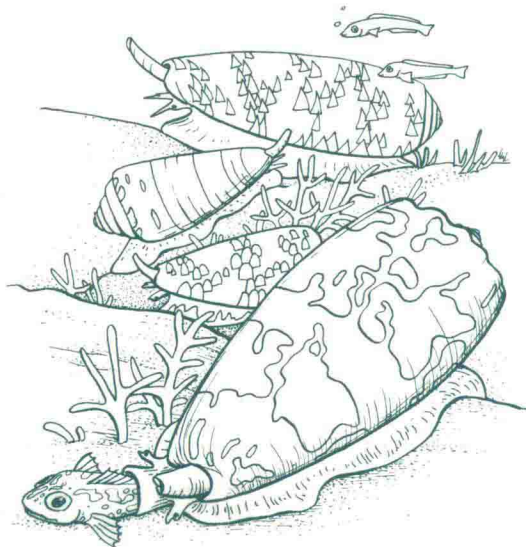
C语言程序设计 ——面向思维的拓展

The C Programming Language

肖乐 主编

董卓莉 王云侠 副主编

- 40余个微课视频，直观解析C语言重难点
- 例题采用“提出问题—思路分析—程序代码—运行结果—说明”的形式
- 课后大量习题，使读者在立体化学习空间中快速掌握所学知识、技能



高校系列



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

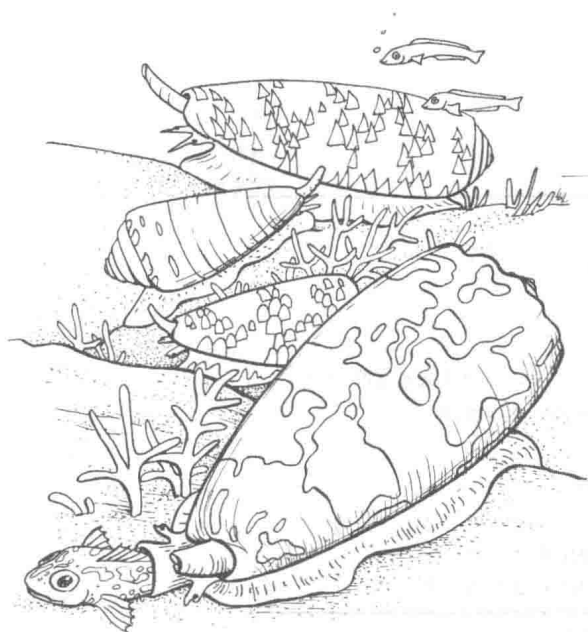
21st Century University Planned Textbooks of Computer Science

C语言程序设计 ——面向思维的拓展

The C Programming Language

肖乐 主编

董卓莉 王云侠 副主编



高校系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C语言程序设计：面向思维的拓展 / 肖乐主编. --
北京：人民邮电出版社，2016.2
21世纪高等学校计算机规划教材. 高校系列
ISBN 978-7-115-41580-6

I. ①C… II. ①肖… III. ①C语言—程序设计—高等
学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第016787号

内 容 提 要

本书是根据我国应用型大学的实际情况，结合当前移动客户端应用情况而编写的实用、立体化教材，全书主要内容包括：引言、算法、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、自定义数据类型和文件等，重点内容附有与网上资源相匹配的二维扫描码。

本书内容丰富、层次清晰、深入浅出、重点突出、图文并茂，所有章节紧扣基本要求，充分突出教材的基础性、应用性和创新性。具体知识讲解全面透彻，例题丰富，每个例题都采用“提出问题—思路分析—程序代码—运行结果—说明”的形式展开，符合读者的认识规律，容易掌握。二维码的设置方便学生使用移动设备随时学习链接内容，结合课后大量习题，使学生在立体化学习空间中快速掌握所学知识、技能。

本书可以作为普通高等学校非计算机专业公共课的教材及全国计算机等级考试参考书，也可以作为自学教材。

◆ 主 编 肖 乐

副 主 编 董卓莉 王云侠

责任编辑 刘 博

责任印制 沈 蓉 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

固安县铭成印刷有限公司印刷

◆ 开本：787×1092 1/16

印张：15.5

2016年2月第1版

字数：407千字

2016年2月河北第1次印刷

定价：39.80元

读者服务热线：(010)81055256 印装质量热线：(010)81055316

反盗版热线：(010)81055315

前言

C 语言是一种经典的计算机语言,也是普通高等学校非计算机专业设置的一门基本计算机课程,它的普及与深入程度将直接影响我国各个行业、领域计算机应用的发展水平,因此,作为非计算机专业学生的程序设计入门课程,C 语言对培养读者的计算机专业素养、兴趣及计算机思维意义重大,需要教材实用并且有特色。本书的目标就是编写易懂、专业、时尚、实用的 C 语言程序设计教材。

本书内容丰富、结构清晰、深入浅出、图文并茂，结合二维码扫描，突出教材的基础性、应用性和创新性。全书主要内容包括：引言、算法、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、结构体和共用体及文件等。

本书编写组成员是多年活跃在大学计算机通识平台课程讲台的教师,并主持和参与多项教改项目及编写多部教材,从教学实践过程中总结经验组织教材内容。非计算机专业学生以提高其计算机操作能力和锻炼计算思维为目的,本书的编写侧重于基本技能和应用能力培养,在知识点的讲授上向应用方面倾斜,每一个例题都按照“提出问题—思路分析—程序代码—运行结果—说明”的步骤展开,符合读者的认识规律,容易掌握,课后附加了大量习题,期望学生做到学以致用,在入门基础上培养自己分析问题、解决问题的能力,提高程序设计能力。每一章内容都力图使读者能够在知识学习过程中,通过知识感悟计算思维;在思维锻炼的过程中,通过思考进一步深刻地理解知识,最后达到融会贯通。

另外,本书还根据教学大纲要求,研究新型教材的编写方法,特别是课程颗粒化粒度的研究,利用教育心理学、课程知识本体理论,重新构建课程知识体系,找到适合计算机通识平台课程的微课程粒度。然后依照颗粒度,作为新教材的章目结构和段落分配依据。在纸质版教材的部分知识点段落,印刷一个与网上资源匹配的二维码,可以及时指导学生利用移动设备进行线上学习。本书除了提供知识链的横向学习模式以外,还提供思维链的纵向思考模式,从而达到计算机技能和计算思维同时锻炼的目的。这种立体化教材的建设,打破了传统纸质书籍的单向传输方式,丰富了学习的互动手段,而且为收集和分析学生教材学习情况提供了技术及数据支撑。

本书全部由河南工业大学的一线教师队伍编写完成,肖乐老师为主编,负责全书的编写规划及二维码和相关慕课视频的制作。具体编写分工如下:第1章由杨晓峰编写,第2章由董卓莉编写,第3章由杨爱梅编写,第4章由周颜编写,第5章由朱红莉编写,第6章由李周芳编写,第7章由李琳编写,第8章由张翼飞编写,第9章由王云侠编写,第10章由艾英山编写。

本书可以作为普通高等学校非计算机专业公共课的教材及全国计算机等级考试参考书,也可以作为自学教材。

编者

2015 年 12 月

目录

第1章 引言	1
1.1 计算机程序	1
1.2 计算机语言	1
1.2.1 机器语言	1
1.2.2 中级语言	2
1.2.3 高级语言	2
1.3 C语言的发展及其特点	2
1.3.1 C语言的发展	2
1.3.2 C语言的特点	3
1.4 最简单的C语言程序	4
1.4.1 最简单的C语言程序举例	4
1.4.2 C语言程序的结构	5
1.5 C语言程序运行的步骤与方法	5
1.6 本章小结	6
习题一	7
第2章 算法	8
2.1 算法概述	8
2.2 简单算法举例	9
2.3 算法的特性	10
2.4 算法描述方法	10
2.4.1 用自然语言描述算法	11
2.4.2 用流程图描述算法	11
2.4.3 3种基本结构和改进的流程图	13
2.4.4 用计算机语言描述算法	16
2.4.5 用伪代码描述算法	17
2.5 本章小结	18
习题二	18
第3章 数据及顺序程序设计	19
3.1 数据的表现形式及其运算	19
3.1.1 常量与变量	19
3.1.2 数据类型	21
3.1.3 整型数据	22
3.1.4 实型数据	25

3.1.5 字符型数据	27
3.1.6 变量的赋值	30
3.1.7 算术运算符与算术表达式	31
3.1.8 逗号运算符和逗号表达式	34
3.1.9 不同类型数据之间的转换	35
3.2 C语句	36
3.2.1 C语句概述	36
3.2.2 最基本的语句——赋值语句	37
3.3 数据的输入输出	40
3.3.1 输入输出概念	40
3.3.2 用printf格式输出函数输出数据	40
3.3.3 用scanf格式输入函数输入数据	46
3.3.4 字符数据的输入输出	50
3.3.5 程序综合实例	51
3.4 本章小结	52
习题三	52
实验 顺序程序设计	54
第4章 选择结构程序设计	56
4.1 用if语句实现选择结构	56
4.1.1 if语句的一般形式	56
4.1.2 使用if语句实现选择结构	58
4.2 if语句中的表达式	61
4.2.1 关系表达式及关系运算符	61
4.2.2 逻辑表达式及逻辑运算符	61
4.2.3 表达式应用举例	61
4.3 条件运算符和条件表达式	62
4.4 选择结构的嵌套	64
4.5 用switch语句实现多分支选择结构	66
4.6 选择结构程序综合实例	69
4.7 本章小结	74
习题四	75
实验 选择结构程序设计	77

第 5 章 循环结构程序设计80	7.2.2 函数调用时的数据传递..... 132
5.1 用 while 语句实现循环.....80	7.2.3 函数的声明 134
5.2 用 do...while 语句实现循环.....82	7.3 函数的嵌套和递归调用 135
5.3 用 for 语句实现循环.....84	7.3.1 函数的嵌套调用 135
5.4 循环的嵌套86	7.3.2 函数的递归调用 136
5.5 break 语句和 continue 语句.....88	7.4 数组作为函数参数..... 138
5.6 循环程序综合实例89	7.4.1 数组元素作函数实参..... 138
5.7 本章小结104	7.4.2 数组名作函数参数 139
习题五105	7.5 变量的作用域和存储类型..... 142
实验 循环结构程序设计110	7.5.1 变量的作用域 142
第 6 章 数组113	7.5.2 变量的存储类型 145
6.1 一维数组的定义和引用113	7.6 本章小结..... 147
6.1.1 一维数组的定义113	习题七..... 147
6.1.2 一维数组元素的引用114	实验 函数..... 148
6.1.3 一维数组的初始化114	第 8 章 指针 150
6.1.4 一维数组程序举例.....115	8.1 指针变量..... 150
6.2 二维数组的定义和引用115	8.1.1 指针的概念 150
6.2.1 怎样定义二维数组116	8.1.2 指针变量的定义与引用..... 152
6.2.2 二维数组元素的引用116	8.1.3 指针变量作为函数参数..... 154
6.2.3 二维数组的初始化116	8.2 通过指针引用数组..... 157
6.2.4 二维数组应用举例.....117	8.2.1 指向数组元素的指针..... 157
6.3 字符数组118	8.2.2 通过指针引用数组元素..... 158
6.3.1 定义字符数组118	8.2.3 用数组名作为函数参数..... 161
6.3.2 字符数组初始化.....119	8.2.4 通过指针引用多维数组..... 167
6.3.3 引用字符数组中的元素119	8.3 通过指针引用字符串..... 173
6.3.4 字符串和字符串结束标志120	8.4 指向函数的指针 176
6.3.5 字符数组的输入输出120	8.4.1 函数指针 176
6.3.6 字符串处理函数.....121	8.4.2 用函数指针变量调用函数..... 176
6.4 本章小结124	8.4.3 定义和使用指向函数的 指针变量 177
习题六124	8.5 本章小结..... 179
实验 数组125	习题八..... 179
第 7 章 函数127	实验 指针..... 182
7.1 函数的定义127	第 9 章 结构体与共用体 185
7.1.1 引例127	9.1 结构体类型引入 185
7.1.2 函数的分类128	9.2 结构体类型与结构体变量..... 186
7.1.3 函数的定义129	9.2.1 结构体类型的声明 186
7.2 函数的调用130	9.2.2 结构类型变量的定义..... 187
7.2.1 调用函数130	9.2.3 结构体变量的初始化和引用..... 188

9.3 结构体数组	191
9.4 结构体指针	193
9.4.1 指向结构变量的指针	193
9.4.2 指向结构数组的指针	195
9.5 结构体与函数	197
9.6 动态数据结构——链表	198
9.6.1 链表的概念	200
9.6.2 链表的操作	201
9.7 共用体类型	209
9.7.1 共同体类型声明	209
9.7.2 引用共用体变量的方式	210
9.7.3 共用体类型数据的特点	210
9.8 枚举类型	213
9.8.1 枚举类型的定义和枚举变量的 说明	213
9.8.2 枚举类型变量的赋值和使用	213
9.9 用 typedef 声明新类型名	215
9.10 本章小结	216
习题九	216
实验 结构体	219

第 10 章 文件	221
10.1 文件概述	221
10.1.1 文件的概念	221
10.1.2 文件的分类	221
10.1.3 文件访问	223
10.2 打开与关闭文件	223
10.2.1 打开数据文件	224
10.2.2 关闭数据文件	225
10.3 顺序读写数据文件	225
10.3.1 向文件读写字符	225
10.3.2 向文件读写字符串	227
10.3.3 格式化的方式读写文件	229
10.3.4 二进制的方式读写文件	230
10.4 随机读写数据文件	232
10.4.1 文件位置标记及其定位	232
10.4.2 随机读写函数	233
10.5 文件读写的出错检测	236
10.6 本章小结	237
习题十	237
实验 文件	239

第1章

引言

1.1 计算机程序

程序，究其本质用一句话来概括，就是一组计算机能识别和执行的指令。

再解释详细一下，计算机程序或者软件程序（通常简称程序）就是指一组指示计算机或其他具有消息处理能力的装置每一步动作的指令，通常用某种程序设计语言编写，运行于某种目标体系结构上。打个比方，一个程序就像一个用汉语（程序设计语言）写下的红烧肉菜谱（程序），用于指导懂汉语和烹饪手法的人（体系结构）来做这个菜。

通俗地讲，程序就像一个“传令官”，将我们的旨意传达给计算机，让计算机去执行。比如，我们要写一篇文章，需要让 Word.exe 这个程序将我们输入的文字记录下来，然后保存或者打印出来；我们想通过计算机跟他人聊天，就需要 QQ.exe 这个程序将我们想说的话通过计算机传递给对方。一个程序，可以接受我们下达的指令，然后让计算机去执行。这就是程序最本质的特征。

1.2 计算机语言

计算机程序设计语言（通常简称编程语言），是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。计算机语言能够让程序员准确地定义计算机所需要使用的数据，并精确地定义在不同情况下应当采取的行动。

自 1946 年计算机问世以来，人们便想利用计算机解决更多的实际问题，改变人类生活，但计算机并不是天生“自动”工作的，要使计算机按照人们的意愿工作，就必须事先告诉它要解决什么样的问题，步骤是什么。那么人和计算机又是如何沟通的呢？这就需要人们以某种程序设计语言为工具对解决问题的步骤进行描述（这种描述就是程序），输入到计算机里用于指挥计算机进行一步步的操作，最后达到解决问题的目的。

计算机程序设计语言的发展一般分为如下 3 个阶段。

1.2.1 机器语言

计算机不懂人类语言，它只能识别由 0 和 1 组成的二进制数字序列。这些序列表示数和指令，

计算机就靠这些二进制的数字序列和人进行交流，我们称这种由 0 和 1 组成的二进制数字序列为机器语言。这些序列的特点是每一步都能操作，所有步骤执行完后对应的问题得到解决。用机器语言编写的程序，计算机能直接执行，但可读性极差且不便移植。机器语言是一种低级语言。

1.2.2 中级语言

为了便于理解和记忆，人们采用一些指令助记符和符号地址来代替机器语言的指令代码，这样就产生了中级语言，其代表就是汇编语言。如用 ADD 代表加法，用 SUB 代表减法。这种指令称为汇编指令，采用汇编指令的语言称为汇编语言。用汇编语言编写的程序，计算机硬件不能直接理解和执行，需要通过一种叫汇编程序的翻译软件将其翻译成机器语言目标程序后，计算机才可以执行。与机器语言相比，汇编语言虽记忆难度有所下降，但不同的计算机其汇编指令也不尽相同，故通用性仍较差。

1.2.3 高级语言



计算机语言的发展

机器语言和汇编语言都是面向机器的语言，它们对计算机硬件的依赖性太强，普通用户很难使用它们编程序，人们希望出现一种既接近自然语言又便于学习和记忆，同时能脱离机型要求的语言。在 1955 年，IBM 公司完成了第一个用英文单词、数学公式按照一定逻辑关系及严格的语法规则构成的程序设计语言，即第一个高级语言——FORTRAN 的最初版本。以后相继涌现出的高级语言版本有 BASIC 语言、Pascal 语言、C 语言等。其中最流行的当数 C 语言，它既可以用来写应用软件，也可写

系统软件（如 UNIX 操作系统）。

1.3 C 语言的发展及其特点

1.3.1 C 语言的发展

C 语言的发展颇为有趣，简单描述如下。

C 语言的原型是 ALGOL 60 语言，也称为 A 语言。

1963 年，剑桥大学将 ALGOL 60 语言发展成为 CPL（Combined Programming Language）语言。

1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，于是产生了 BCPL 语言。

1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改。为它起了一个有趣的名字“B 语言”，意思是将 CPL 语言“煮干”，提炼出它的精华；并且还用了 B 语言写了第一个 UNIX 操作系统。

1973 年，B 语言也被人“煮”了一下。美国贝尔实验室的 D.M.RITCHIE 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二字母作为这种语言的名字，这就是 C 语言。

为了使 UNIX 操作系统得到推广，1977 年 Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。

1978年, Brian W.Kernighian 和 Dennis M.Ritchie 出版了著作《The C Programming Language》, 从而使 C 语言成为目前世界上流行最广泛的高级程序设计语言。

1988年, 随着微型计算机的普及, C 语言出现了许多版本。由于没有统一的标准, 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, 美国国家标准研究所 (ANSI) 为 C 语言制定了一套 ANSI 标准。1989年, ANSI 公布了一个完整的 C 语言标准 ANSI X3.159-1989 (即 ANSI C 或 C89)。

1990年, 国际化标准组织 ISO 接受 C89 作为国际标准 ISO/IEC9899:1990。

1999年, ISO 对 C90 做了一些修订, 在基本保留原来的 C 语言特征的基础上, 针对应用的需要, 增加了一些功能, 尤其是 C++ 中的一些功能, 命名为 ISO/IEC9899:1999。

1.3.2 C 语言的特点

C 语言作为使用广泛的一种编程语言, 具有以下特点。

1. 简洁紧凑, 灵活方便

C 语言一共只有 32 个关键字, 9 种控制语句, 程序书写自由, 主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作, 而这三者是计算机最基本的工作单元。

2. 运算符丰富

C 语言的运算符包含的范围很广泛, 共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 语言的运算类型极其丰富, 且表达式类型多样化, 灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据结构丰富

C 语言的数据类型有: 整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等, 能用来实现各种复杂的数据类型的运算。并引入了指针概念, 使程序效率更高。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。且计算功能、逻辑判断功能强大。

4. C 语言是结构式语言

结构式语言的显著特点是代码及数据的分隔化, 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、维护以及调试。C 语言是以函数形式提供给用户的, 这些函数可方便地调用, 并具有多种循环、条件语句控制程序流向, 从而使程序完全结构化。

5. C 语言语法限制不太严格, 程序设计自由度大

一般的高级语言语法检查比较严, 能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。

6. C 语言允许直接访问物理地址, 可以直接对硬件进行操作

C 语言既具有高级语言的功能, 又具有低级语言的许多功能, 能够像汇编语言一样对位、字节和地址进行操作, 而这三者是计算机最基本的工作单元, 可以用来写系统软件。

7. C 语言程序生成代码质量高, 程序执行效率高

一般只比汇编程序生成的目标代码效率低 10%~20%。

8. C 语言适用范围大, 可移植性好

C 语言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。

1.4 最简单的 C 语言程序

1.4.1 最简单的 C 语言程序举例

相对于其他的语言来说，C 语言的知识点繁多且琐碎，为了便于同学们尽快入门，在此先举一个简单的例子。

【例 1.1】编写程序，输出以下信息。

"I am a student. "

思路分析如下。

在主函数中调用库函数 printf() 直接输出。

程序代码如下。

```
#include<stdio.h>           //编译预处理指令
int main()                  //定义主函数
{                            //函数开始标志
printf("I am a student.\n"); //调用函数输出内容
return 0;                   //返回函数值
}                             //函数结束标志
```

运行结果如下。

```
I am a student.
Press any key to continue
```

说明如下。

以上运行结果是在 Visual C++ 6.0 环境下运行程序时得到的。其中，第一行是程序的运行结果，第二行是 Visual C++ 6.0 系统在输出完成后自动输出的信息，意思是“按任意键进行下一步”。当用户在此状态下按任意键，则运行结果窗口自动关闭并返回程序窗口，以便于用户进行下一步操作。

程序分析如下。

第一行。stdio.h 是系统提供的一个文件，是“standard input & output”即“标准输入输出”的缩写，后缀.h 表明该文件类型为头文件（header file）。程序第四行调用了输出函数，编译系统要求程序事先提供关于该函数的相关信息，这些相关信息都包含在 stdio.h 文件中，因此在第一行用 #include 编译预处理指令将该头文件引入（注意：因为每个程序都必须有输出，所以每个程序都必含有该语句行，建议始终写在程序的第一行）。

第二行。定义主函数 main()。main() 是主函数名，每个程序都必须有且仅有一个主函数，程序从主函数开始，在主函数中结束。int 是整型 integer 的缩写，在此表示定义主函数为整型，也意味着函数结束后得到的返回值为整型。

第三行。“{”和“}”是函数开始与结束的标识符。

第四行。调用格式输出函数 print() 输出结果。print() 是 C 编译系统提供的一个库函数，行中双引号内的字符原样输出，“\n”是一种转义字符，作用是在当前内容输出结束后换行，将当前输

出位置跳到下一行行首。

第五行。当程序结束时返回值 0 到调用函数处。

另，整段程序右侧“//”是注释的标志，在其后的内容为注释，对运行不起作用。

1.4.2 C 语言程序的结构

C 语言程序结果有如下特点。

- (1) 一个程序由一个或若干个源程序文件组成。
- (2) 函数是程序的主要组成部分。
- (3) 程序有且仅有一个 `main()` 函数，程序运行总是从 `main()` 函数开始执行，并在 `main()` 函数中结束。
- (4) 程序中的所有操作都是由函数中的语句实现的，“;”是语句的结束标志。

1.5 C 语言程序运行的步骤与方法

遇到一个问题，在分析问题、设计算法、画流程图并编写代码之后，我们得到的只是 C 语言的源程序。这种源程序是不能被计算机直接识别的，我们还需要经过编译、连接而形成可执行的目标程序才能运行调试，并最终得到运行结果。具体来说，运行 C 语言程序分以下几个阶段。

(1) 录入并编辑代码。打开 C 语言开发环境，通过键盘录入代码，在经过核对无误后将程序存入用户指定文件夹（若不指定，则自动存入用户当前文件夹），保存后得到源程序文件 `***.c`。

(2) 编译源程序。先用 C 编译系统提供的预处理器对程序中的预处理指令进行编译预处理，比如对 `stdio.h` 或 `math.h` 等头文件中的内容读入并代替原程序行，再将读入信息与其他部分组成一个完整的源程序文件进行正式编译。编译的主要作用就是检查语法错误，经过调试——修改——调试过程后，编译系统会自动将源程序转换为二进制形式的目标程序 `***.obj`。

(3) 链接处理。将所有经过编译处理而得到的目标程序以及函数库连接起来，生成一个可供计算机执行的可执行程序 `***.exe`。

(4) 运行程序，得到结果。

程序运行流程图如图 1-1 所示。

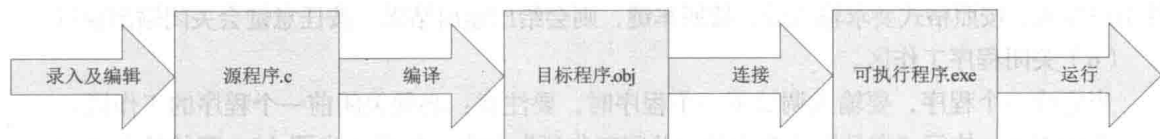


图 1-1 程序运行流程图



编译、连接、运行过程

开发环境种类很多，本书选用用户最广泛的 Microsoft Visual C++ 6.0 作为开发环境。具体操作规程如下。

(1) 进入和退出 Visual C++ 集成开发环境。

启动并进入 Visual C++ 集成开发环境的方法如下。

① 选择“开始”菜单中的“程序”，然后选择“Microsoft Visual Studio 6.0”级联菜单，再

选择“Microsoft Visual C++ 6.0”。若桌面上创建有 Microsoft Visual C++6.0 的快捷方式，直接双击该图标。

② 如果已经创建了某个 Visual C++工程，双击该工程的.dsw (Develop Studio Workshop) 文件图标，也可进入集成开发环境，并打开该工程。

退出 Visual C++集成开发环境的方法：选择 File|Exit 菜单，可退出集成开发环境。

(2) 创建一个 C 语言程序文件。

进入 Visual C++集成开发环境后，选择 File|New 菜单，弹出“New”对话框，单击“Files”标签，打开其选项卡，在其左边的列表框中选择“C++ Source File”项，单击“确定”按钮，即可新建一个文件。此时在编辑窗口中输入你自己的程序。

(3) 保存程序。

当输入结束后，执行“文件”菜单中的“保存”命令将文件存盘。建议大家在桌面上用自己的英文名字创建一个新的目录，然后将 C 语言文件存在这个新建目录中。



在保存文件时，应设置扩展名为“.c”，否则，系统将按 C++扩展名“.cpp”保存。

(4) 编译调试程序。

执行“编译”菜单中的“构件 (build)”命令。在编译过程中，Visual C++会生成一个同名的工作区 (workspace)。

如果程序没有错误，那么在正下方的信息窗口中会显示如下内容。

-0 error(s), 0 warning(s)

有时会出现几个警告性信息 (warning)，不会影响程序的执行。

如果程序中有多多个致命性错误 (error)，那么会提示如下信息。

-N error(s), M warning(s)

此时，拖动信息窗口右侧的滚动条，可以看到所有编译器给出的错误提示，包括错误是在第几行以及是什么错误。双击某行出错信息，程序窗口中会用箭头指示对应出错位置。根据信息窗口中的错误提示，修正错误。修正好后，保存，重新编译，直至无错误为止。

(5) 执行程序。

执行“编译”菜单中的“执行 (Execute)”命令，执行程序。当运行 C 语言程序后，VC++将自动弹出数据输入输出窗口。若程序中有 scanf 语句或者 getchar 语句等输入，那么窗口会停住，等待用户输入，按照格式要求输入后，按回车键，则会给出输出结果。按任意键会关闭输出窗口。

(6) 关闭程序工作区。

当完成一个程序，要输入调试下一个程序时，要注意：必须关闭前一个程序的工作区。

关闭方法：执行“文件”菜单中的“关闭工作区”命令，然后从步骤 (2) 开始输入调试下一个程序。

1.6 本章小结

本章简要介绍了计算机程序和计算机语言的概念，简述了 C 语言的特点及发展过程，举例详

细说明了C语言的上机调试过程及运行过程。通过本章的学习,使得读者可以掌握C语言的初步知识。

习 题 一

一、填空题

1. 调试一个C语言程序要经过____、____、____、____4个步骤。
2. 在C语言中,主函数名是____,一个C语言程序有____个主函数。
3. 在C语言中,头文件的扩展名为____。
4. 在C语言中,语句结束的标志是____。

二、简答题

1. 什么是计算机程序?
2. 什么是计算机语言?
3. C语言的特点主要有哪些?



第2章 算法

2.1 算法概述

在中国,“算法”一词最早出现在《周髀算经》和《九章算术》等古代文献中。在《九章算术》中,给出了四则运算、最大公约数、最小公倍数、开平方根、开立方根和线性方程组求解等的算法。此外,三国时期的刘徽是中外最早给出求圆周率算法的数学家之一,他提出了有名的圆周率算法,即“刘徽割圆术”。



刘徽割圆术

在西方,算法(algorithm)一词原为“algorism”,由9世纪提出“算法”这个概念的波斯数学家花拉子米的名字(“al-Khwarizmi”)音译而来,意思是“花拉子米”的运算法则,在18世纪演变为“algorithm”。

在《韦氏大学词典(第九版)》中,“算法”解释为“求解数学问题的一个过程,该过程步骤有限,通常还涉及重复的操作;广义地说,算法是按部就班解决一个问题或完成某个目标的过程。”

几乎所有的现实人类活动,都必须遵循事先安排的计划步骤来进行。

以披萨的制作为例,硬件(hardware)包括烤箱、饼坯、奶酪蔬菜等各式配料,软件(software)一般指披萨的制作步骤说明。这里,披萨的制作步骤说明即我们所指的算法(algorithm),算法的输入(input)为饼坯、配料等食材,输出(output)则为披萨。烤箱能够完成的工作包括开和关烤箱门、旋转开关旋钮以及加热等,配合以正确的披萨制作步骤说明并照此操作,就能够完成从基本食材到美味的披萨的过程。

和人类活动的计划步骤类似,算法,或者计算机算法,特指解决数学问题中的方法步骤。众所周知,数据在计算机中以二进制形式参与运算,计算机也仅仅能够执行二进制位的翻转(将0置1、将1置0)、置0或位测试等简单的位操作。计算机是如何把普通的对位的操作转换成我们日常见到的由计算机完成的各项应用的呢?对计算机操作的描述就是算法的研究内容。

算法和程序是什么关系呢?如同披萨的制作一样,不仅要有对制作步骤的说明,也要有对食材的说明。在程序设计中,既要有对操作的描述(称为“算法”),也要有对数据的描述(称为“数据结构”),著名的计算机科学家沃斯(Niklaus Wirth)提出了如下著名的公式:

程序=算法+数据结构

可见,参与运算的数据是基础,是程序中指令操作的对象,而算法是程序设计的灵魂。

按照参与运算的对象不同,计算机算法可以分为数值运算算法和非数值运算算法两类。

数值运算算法旨在求取数值解,比如圆周率 π 值的求解,线性方程组的求解,计算矩阵的特征值,求解数值积分等。对于绝大多数的数值运算而言,都有成熟的算法以及相应的程序供人们选择使用。

而基于比较关系运算的算法称为非数值运算算法,常见的有各种排序算法(比如对一个专业或年级的学生的成绩进行排序)、搜索算法(查找平均分大于90分的所有学生)等。非数值运算中,只有经典问题有较为成熟的算法,许多问题可以重新设计新的算法,这对程序设计人员是一项有趣的挑战。

判断一个运算算法的优劣主要从三个方面进行考察,即误差分析、复杂度分析和稳定性分析,好的算法误差小、耗时少且抗干扰(或称为鲁棒性好)。

值得一提的是,欧几里得算法(又称“辗转相除法”,用于计算两个整数 m 和 n 的最大公约数)被人们公认为历史上的第一个算法。

2.2 简单算法举例

【例 2.1】计算两个整数的最大公约数。

定义:依据对最大公约数的定义,两个不全为0的非负整数 m 和 n 的最大公约数记为 $\gcd(m,n)$,代表能够整除(即余数为0) m 和 n 的最大正整数。

思路分析:以系统论述几何学而著称的《几何原本》(欧几里得,公元前3世纪)的第VII卷中简要描述了最大公约数算法。用现代数学的语言可将该算法描述为:重复应用下列等式直至 $m \bmod n=0$ 。

$$\gcd(m,n)=\gcd(n,m \bmod n) \quad (m \bmod n \text{ 表示 } m \text{ 除以 } n \text{ 之后的余数})$$

由于 $\gcd(m,0)=m$, m 最后的取值也就是 m 和 n 初值的最大公约数。

比如,依照上述算法,72和54的最大公约数 $\gcd(72,54)$ 可以写为

$$\gcd(72,54)=\gcd(54,18)=\gcd(18,0)=18$$

算法描述:上述算法就是著名的欧几里得算法,用结构化的语言可以描述如下。

第一步:如果 $n=0$,那么将 m 的值作为结果返回值,算法结束;否则(即 $n \neq 0$),算法进入第二步。

第二步: m 除以 n ,得到余数 r 。

第三步:分别将 n 和 r 作为新的 m 和 n ,重新执行第一步。

【例 2.2】求 $1+2+3+\cdots+100$,即

$$\sum_{n=1}^{100} n$$

思路分析:这是一个累加问题。

算法一:在数学上有固定的公式可以使用,即

$$\sum_{n=1}^{100} n = \frac{n(n+1)}{2}$$

因此,可以直接利用表达式 $\text{sum}=n*(n+1)/2$ 来进行求解。

算法二:算法一虽然看上去简单直观,但是由于计算机执行加法的速度达到万亿次/秒的级别,



欧几里得算法

因此执行重复累加求和的方式，从时间效率上来看，甚至优于上述包含乘法的算法。

算法描述：算法二以形式化语言可以描述如下。

第一步：将和 sum 初值置 0，加数 i 初值置 1。

第二步：如果 $i > 100$ ，算法结束；否则（即 $i \leq 100$ ），算法进入第三步。

第三步：将 i 加到 sum 上，之后让 i 的值增加 1，转到第二步。

【例 2.3】求 200 以内的全部素数。

思路分析：（基本试除法）让每个整数 n 被 i 除（ i 的值从 2 变到 $n-1$ ）如果 n 能被 $2 \sim (n-1)$ 之中的任何一个整数整除，则表示 n 肯定不是素数，不必再继续被后面的整数除。

算法描述：求素数的算法可以用形式化的语言描述如下。

第一步：将待判断的整数 n 的初值置为 3，除数 i 的初值置为 2。

第二步：如果 $n > 200$ ，则算法结束；否则，尝试 n 除以 i 。

第三步：如果 $n \bmod i \neq 0$ ，则将 i 增加 1。

第四步：重复执行第三步，再次尝试 n 除以 i ，直到 $n \bmod i \neq 0$ ，将 n 增加 1，回到第二步。

2.3 算法的特性

正如写文章一样，针对要表达的同样的核心思想，可以使用中文表述，也可以使用英语、法语、日语等其他语种。在使用计算机解决实际问题时，程序设计语言只是载体，方案的执行步骤（算法）才是核心。通常，编写程序之前要设计算法。一般情况下，具备以下 5 个基本特性的设计方案才能称之为算法。

（1）有穷性。算法经过许多指令有限次数的执行后会终止。

（2）确定性。算法每一步执行的中间结果都是唯一的，只取决于输入和前面步骤的结果。

（3）可行性。算法在原则上能精确进行，可通过已实现的基本运算执行有限次而完成。

（4）输入。算法接受多个或者 0 个输入。所谓输入是指从外界向计算机输送必要的信息，大部分情况下需要输入，但是有些情况并不需要。

（5）输出。算法有一个或者多个输出。算法的目的是求得问题的“解”。算法的输出不代表计算机一定要有打印输出和屏幕输出，只要获得算法的解，就是算法的输出。

我们结合欧几里得算法对算法特性加以说明。在例 2.1 中，由于每一步的余数都在逐渐减小并且不为负数，必然存在第 N 步时余数为 0，因此算法在有限次执行之后结束。并且，两个不全为 0 的正整数求余的结果是唯一的，只取决于初始输入的 m 和 n 的值和每一步中间步骤得到的 n 和 r 的值。求余运算是基本的算术运算，每一步都能够精确执行。本算法接受输入的两个值 m 和 n ，执行结果为输出最后一次的 m 值。

2.4 算法描述方法

由于算法是对一个问题求解过程的描述，或者说算法是求解问题的基本指令的集合，这些指令规定了所要执行的基本行为，因此，安排执行基本指令的次序至关重要。通常，规定执行基本指令次序的工作是借助于称作“控制结构”的指令的不同组合来完成的。从语义及逻辑含义上讲，