



普通高等教育“十一五”国家级规划教材

张长海 赵大鹏 陈娟 编著

大学计算机程序 设计基础 (C语言)

21世纪
计算机
科学
与
技术
实践
型
教
程

丛书主编
陈明



清华大学出版社



普通高等教育“十一五”国家级规划教材

张长海 赵大鹏 陈娟 编著

大学计算机程序 设计基础(C语言)

21世纪
计算机
科学
与
技术
实
践
型
教
程

丛书主编
陈明

清华大学出版社
北京



内 容 简 介

本书以 C 语言为载体,引进 PAD 图表示程序逻辑,阐述基本的程序设计方法。全书共分 13 章,主要内容包括算法、程序设计方法、函数、数据的组织、程序开发。每章都包含大量例题和习题。

本书最大的特点是以程序设计为主线、以案例为驱动。全书自始至终围绕“程序设计”来讲解,而不是“语言”的简单介绍,摒弃了目前各种程序设计书中流行的“解释程序设计语言”的做法。本书的目的是教会读者怎样编程序,提高读者的程序设计能力,彻底改变“学了程序设计而不会编程序”的现状。

本书整体结构好、图文并茂,力求体现“结构化程序设计”思想,注重培养和训练读者良好的程序设计风格。

本书可以作为面向应用的高等院校中计算机类各个专业程序设计课程、一般高等院校理工科各公共计算机基础课程“高级语言程序设计”、“程序设计基础”、“C 程序设计”、“C 语言”等的教材和参考书;还可以供从事计算机工作的有关人员参考。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

大学计算机程序设计基础: C 语言/张长海,赵大鹏,陈娟编著. —北京:清华大学出版社,2009.5

(21 世纪计算机科学与技术实践型教程)

ISBN 978-7-302-19436-1

I. 大… II. ①张… ②赵… ③陈… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核子(2009)第 013040 号

责任编辑:谢琛 顾冰

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京嘉实印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:17.75

字 数:411 千字

版 次:2009 年 5 月第 1 版

印 次:2009 年 5 月第 1 次印刷

印 数:1~5000

定 价:25.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:032096-01

《21 世纪计算机科学与技术实践型教程》

序

21 世纪影响世界的三大关键技术是：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制定科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21 世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21 世纪计算机科学与技术实践型教程》就是为了填补这部分鸿沟。将理论与实际联系起来，结合起来，使学生不仅学会了计算机科学理论，而且也学会应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，把他们积累的经验、知识、智慧、素质融合于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编 陈明

2005 年 1 月于北京

前 言

随着计算机技术的不断发展,计算机应用的逐步普及,人们对于大学程序设计课的认识也在不断的更新。

最早在 20 世纪 70 年代及其以前,计算机应用十分狭窄,所谓“搞计算机”的人也很少。那时的计算机专业是在研究计算机本身,相应的第一个程序设计课称为“算法语言”。学习该课程的目的是学习程序设计语言本身(绝大部分学校都选讲 ALGOL60),而对于怎么用这个语言不太关心。因为那时的教学目的是让学生研究计算机本身,是要制造计算机,是要研究透程序设计语言本身的语法和语义,并实现它。

进入 21 世纪之后,计算机已经渗透到各个领域,甚至已经普及到家庭,它像电视、冰箱一样,是家用电器之一。计算机专业的规模与 30 年前大不一样,据 2005 年统计,全国有“计算机科学与技术”专业的院校有 741 所,“计算机科学与技术”专业在校生人数超过 45 万;目前据保守估计,全国有“计算机科学与技术”专业的院校超过 800 所,“计算机科学与技术”专业在校生人数超过 50 万。若再加上软件工程、计算机网络、信息技术等计算机类的专业,全国的计算机专业数和在校生数还将比 800 所和 50 万大得多。

面对如此庞大的队伍,我们的毕业生不可能都去研究计算机本身,社会也不需要这么多人从事计算机研究。所以现在计算机专业的含义已经完全与 30 年前的计算机专业不同了,现在绝大部分计算机专业人员主要是做“计算机应用”,甚至是在“应用计算机”,而不是在研究计算机本身了。

由于计算机的普及,高等院校各专业都在使用计算机解决本领域的问题,各专业的学生都需要学习“程序设计”,显然这些非计算机专业人员学习的目的更是在“应用计算机”。

在这种形式下,程序设计课怎么讲?摆在每个组织计算机教学的人面前,更摆在教授程序设计课的教师面前。目前学习程序设计的人大致可以分成三类:

- 从事计算机研究的人员,人数极少;
- 计算机应用和应用计算机的人员,这类人员是绝大多数;
- 程序编码人员,这类人员是高职高专培养的目标,社会需求极大,但学校培养的人数有限。

本书针对计算机应用人员和非计算机专业人员(应用计算机人员)。在面向应用的大背景下,程序设计课不应该再去讲程序设计语言本身了,而应该讲程序设计语言的应用。应该把授课重点从讲授语言的符号、语句等语言成分上,转移到讲授程序设计上。教学的目的是教会学生“怎么编程序”,而不是背几个语言符号、说明、语句等。应该把该课程的

名字从“×××语言”改为“高级语言程序设计”、“程序设计基础”或“程序设计”。尤其一些非计算机专业开设该课程,更应该跳出讲授“语言”的误区。因为他们更是在“应用计算机”,是用计算机解决本专业领域的问题,更没有必要背那些死的语言符号、语法概念,更应该是学会怎么编程序。

本书秉承面向应用的主导思想,依托 C 语言讲述“程序设计”,重点在于程序设计。在这种思想指导下,对 C 语言本身采取“有所取、有所不取”的策略。对于那些常用的语言成分,直接与讲述程序设计方法有关的语言成分,穿插在程序设计过程中详细准确的介绍;对于那些与程序设计方法联系不太紧要,但是还使用的部分,放在最后简单介绍;而对于那些与讲述程序设计方法关系不太大,也不常用的部分则根本不涉及。

本书秉承“授之以渔”而非“授之以鱼”的理念组织教学内容。以案例为驱动,使用大量例题讲解程序设计思想和方法。案例不是为了解释语言概念,而是从构造算法出发,以训练学生的实际编程能力为目标。彻底改变过去那种单纯解释语法、语义等语言成分的做法。改变那种提出一个很小的问题,然后给定相应的不太大的程序,最后解释程序中各个语句、说明的做法。让程序设计始终贯穿于整个教学过程,使教学内容更贴近应用。针对程序设计的每个知识模块都采取如下模式讲授:

提出有意义的问题→设计算法→分析算法特点→编出程序

→介绍使用的 C 语言成分→配合讲述大量例题→课后习题与实践

在上述过程中,把重点放在设计算法和讲述算法特点上。例如,全书开篇从有趣的“鸡兔同笼”问题开始,引进算法、程序、程序设计等概念,即讲授了抽象的概念又调动起了学生的学习兴趣。

作为大学本科计算机基础课教材,本书具有如下特点:

(1) 最大的特点是以程序设计为主线、以案例为驱动。按程序设计的思路组织全书内容,真正的在讲授程序设计,而不是讲语言,摒弃了目前各种程序设计书中流行的主要“解释程序设计语言”的做法。

(2) 整体结构好,章节安排合理,由浅入深地介绍程序设计知识。例如有关函数的知识,由浅入深的分散到 4 章中介绍;有关指针的知识也分散到 5 章中介绍。免得集中在一章,使读者学起来枯燥乏味,接受困难。

(3) 全书自始至终贯穿结构化程序设计思想,所有例题都具有良好的结构和程序设计风格。目的是给读者一个示范,使读者从开始学习程序设计就养成一个良好的程序设计习惯和风格。

(4) 图文并茂,引进 PAD 图表示程序逻辑。PAD 图的结构比传统的流程图、NS 图等都好,同时也比直接用程序表示算法更直观,易于理解。

(5) 配备大量例题和习题,并且全部为程序设计题目。例题讲解从构造算法出发,以训练读者的编程能力为目标;概念、语言成分的介绍穿插在程序设计之中。本书中的全部例题都在 Microsoft Visual C++ 6.0 的环境下调试通过。大量的习题供读者做练习和进一步提高使用。

全书共 13 章,大致分为四部分。

第一部分基本知识,为第 1 章,介绍计算、算法和程序设计基本概念。

第二部分程序设计,包括第2~5、9、11章。第2章简单程序设计,介绍顺序程序设计、数据及其类型、表达式、赋值、I/O;第3章介绍分支程序设计;第4章介绍循环程序设计;第5章简单介绍模块化程序设计思想,引进子程序和函数概念;第9章进一步介绍函数,讲述参数、作用域、递归程序设计;第11章介绍程序开发和结构化程序设计,包括结构化程序设计原则、程序风格、自顶向下逐步求精的程序设计技术。

第三部分数据组织,包括第6~8、10、12章。第6章讲述批量数据组织——数组;第7章介绍指针;第8章讲述对复杂的表单数据的描述,引进结构体;第10章讲述外部数据组织——文件及其操作;第12章讲述动态数据组织及其在程序设计中的应用。

第四部分为第13章,若干深入的问题。进一步介绍函数,讲述函数作参数、函数值、函数副作用、间接递归等;以及存储类别、位操作、位段、goto、编译预处理等。

本书第1~4章由吉林大学张长海执笔,第5、6、8~11章由长春税务学院赵大朋执笔,第7、12、13章由吉林大学陈娟执笔。最后由张长海统稿。

在本书的编写过程中作者参阅并引用了国内外诸多同行的文章、著作,在此作者向他们致意,并恕不一一列举、标明。在本书的成书和出版过程中得到清华大学出版社的帮助和大力支持,作者在此表示由衷的感谢。

限于作者学术水平有限,错误和不足在所难免,敬请各位读者批评指正。作者十分感谢。

作者

2009年3月于长春

目 录

第 1 章 绪论	1
1.1 “鸡兔同笼”——计算	1
1.2 算法	2
1.3 PAD 图	3
1.4 程序	4
1.5 运行	5
1.6 计算机的基本结构	7
1.7 C 语言	7
1.8 Visual C++ 集成开发环境	9
1.8.1 启动 VC++	9
1.8.2 独立文件模式下建立环境	9
1.8.3 独立文件模式下录入、编辑源程序	9
1.8.4 编译	11
1.8.5 连接与运行	11
1.8.6 程序参数	12
1.8.7 项目管理模式下建立运行环境	13
1.8.8 项目管理模式下录入、编辑源程序	14
本章小结	16
习题一	16
第 2 章 简单程序	18
2.1 行程问题——简单程序	18
2.2 语句	19
2.3 表达式	20
2.3.1 表达式语句	22
2.3.2 赋值	22
2.4 基本符号	23
2.4.1 关键字	23

2.4.2	标识符	23
2.4.3	间隔符	24
2.4.4	注释	25
2.5	数据及其类型	26
2.5.1	浮点类型	26
2.5.2	整数类型	27
2.5.3	字符类型	27
2.5.4	混合运算	28
2.6	常量和变量	28
2.6.1	字面常量	28
2.6.2	常量标识符	31
2.6.3	变量	31
2.7	类型转换	32
2.8	输入输出	34
2.8.1	字符输入	34
2.8.2	字符输出	35
2.8.3	格式输入	35
2.8.4	格式输出	36
2.9	顺序控制结构	39
	本章小结	43
	习题二	44
第 3 章	分支程序设计	46
3.1	判断成绩是否及格——双分支程序设计	46
3.2	成绩加上获奖信息——单分支程序设计	48
3.3	逻辑判断——布尔类型	54
3.3.1	关系运算	54
3.3.2	布尔运算	55
3.4	获奖分等级——多分支程序设计	55
3.5	表示汽车种类——枚举类型	59
	本章小结	61
	习题三	61
第 4 章	循环程序设计	64
4.1	计算平均成绩——循环程序	64
4.1.1	后判断条件的循环	65
4.1.2	先判断条件的循环	69
4.1.3	for 语句	73

4.2 打印 99 表——多重循环	75
4.3 程序设计实例	79
本章小结	83
习题四	83
第 5 章 模块化程序设计——函数	86
5.1 求给定三角形的重心——模块化程序设计	86
5.2 函数	89
5.2.1 函数定义	90
5.2.2 函数调用	91
5.2.3 先调用后定义	93
5.3 程序设计实例	94
本章小结	97
习题五	97
第 6 章 批量数据组织——数组	99
6.1 成绩统计——数组类型	99
6.1.1 数组声明	100
6.1.2 下标表达式	101
6.1.3 应注意的问题	101
6.2 统计多科成绩——多维数组	102
6.3 程序设计实例	104
6.4 成绩排序——数组初值	113
6.5 输出回文字——字符串	115
6.6 类型定义	119
本章小结	120
习题六	121
第 7 章 指针	124
7.1 指针与变量	124
7.1.1 指针类型和指针变量	125
7.1.2 指针所指变量	126
7.1.3 空指针、无效指针	128
7.2 指针运算	129
7.3 指针与数组	131
7.3.1 用指针标识数组	131
7.3.2 指针数组	132
7.4 指针与字符串	134

7.5 指向指针的指针	137
本章小结	139
习题七	139
第 8 章 表单数据组织——结构体	141
8.1 图书卡管理	141
8.2 图书卡保存——结构体	144
8.2.1 定义结构体类型	145
8.2.2 结构体类型名	146
8.2.3 结构体变量	147
8.2.4 指向结构体变量的指针	148
8.2.5 访问结构体变量的成分	148
本章小结	150
习题八	150
第 9 章 再论函数	153
9.1 参数	153
9.1.1 传递直线方程系数——指针作参数	153
9.1.2 对任意数组排序——数组作参数	155
9.1.3 图书卡检索——结构体作参数	159
9.2 函数值	160
9.2.1 打印月份名——返回指针值的函数	160
9.2.2 读入图书卡片——返回结构体值的函数	162
9.3 作用域	163
9.4 局部量和全局量	164
9.5 计算 $n!$ ——递归程序设计	166
本章小结	171
习题九	171
第 10 章 外部数据组织——文件	174
10.1 重新考虑图书卡管理问题——文件	174
10.2 文件概述	176
10.3 打开和关闭文件	179
10.4 程序参数	180
10.5 字符读写	183
10.6 字符串读写	184
10.7 格式化读写	185
10.8 数据块读写	188

10.9 文件定位	191
本章小结	192
习题十	193
第 11 章 程序开发	195
11.1 求三角形外心——自顶向下、逐步求精	195
11.2 结构化程序设计原则	200
11.3 程序风格	201
11.3.1 行文格式	202
11.3.2 标识符	203
11.3.3 注释	203
11.3.4 对程序说明的建议	203
本章小结	204
习题十一	204
第 12 章 动态数据组织	206
12.1 打印法雷序列——动态数据结构	206
12.2 动态变量	210
12.3 链表	212
12.4 程序设计实例	216
本章小结	220
习题十二	220
第 13 章 若干深入问题	222
13.1 函数	222
13.1.1 不定方向的数组排序——函数指针	222
13.1.2 计算定积分——函数作参数	224
13.1.3 计算算术表达式的值——间接递归	228
13.1.4 函数副作用	230
13.2 运算	231
13.2.1 赋值运算	232
13.2.2 顺序表达式	232
13.2.3 条件表达式	232
13.2.4 位运算	233
13.3 语句	235
13.3.1 break	235
13.3.2 continue	236
13.3.3 for 的延伸	237

13.3.4 goto 和标号	237
13.4 数据组织	238
13.4.1 多维数组与指针	238
13.4.2 位段	241
13.4.3 职工登记卡——共用体	242
13.5 存储类别	245
13.5.1 数据在内存中的存储	246
13.5.2 自动存储类别	246
13.5.3 寄存器存储类别	247
13.5.4 变量的静态存储类别	248
13.5.5 变量的外部存储类别	250
13.5.6 函数的存储类别	250
13.5.7 类型定义符	251
13.6 编译预处理	252
13.6.1 宏定义	252
13.6.2 文件包含	253
13.6.3 条件编译	254
本章小结	256
习题十三	256
附录 A ASCII 字符集	259
附录 B 标准库头文件表	262
参考文献	263

第 1 章 绪 论

现代计算机从出现至今不过 60 年时间,但其发展速度是任何一种新技术都不可比拟的,目前计算机已经渗透到各个领域。本书将以 C 语言为载体向大家介绍如何编制计算机程序,即如何使用计算机解决科技、生产、事务处理等方面的问题。介绍程序设计的基本方法、技术和技巧。

1.1 “鸡兔同笼”——计算

【例 1.1】 我国古代数学著作《孙子算经》中所载“鸡兔同笼”问题如下:“今有鸡兔同笼,上有三十五头,下有九十四足,问鸡兔各几何?”。

解: 怎样解决这个问题? 分析如下:

第一步,解决这个问题应该首先把问题数学化。根据题目条件,设有 x 只鸡, y 只兔,可以列出二元一次方程组:

$$x + y = 35 \quad (1)$$

$$2x + 4y = 94 \quad (2)$$

第二步,解该方程组。解二元一次方程组有各种方法,现在选择消元法。

第三步,使用消元法解上述二元一次方程组,按如下步骤进行:

① 把方程(1)乘以 2: 计算 1×2 ; 35×2 ; 得到方程:

$$2 \times x + 2 \times y = 70 \quad (3)$$

② 把方程(2)减去方程(3): 计算 $2 - 2$, $4 - 2$, $94 - 70$, 得到方程:

$$2 \times y = 14 \quad (4)$$

③ 解该一元一次方程: 计算 $14/2 \rightarrow y$, 得到未知数 y 的值。

④ 把 y 值代入方程(1)得一元一次方程(5):

$$x + 7 = 35 \quad (5)$$

⑤ 解该一元一次方程: 计算 $35 - 7 \rightarrow x$ 得到未知数 x 的值。

如上已经把解决该问题的各个步骤分析清楚。实际计算过程应该是:

① $1 \times 2 \rightarrow a$;

② $35 \times 2 \rightarrow b$;

- ③ $4 - a \rightarrow a$;
- ④ $94 - b \rightarrow b$;
- ⑤ $b/a \rightarrow y$;
- ⑥ $35 - y \rightarrow x$ 。

选择一种程序设计语言,把上述计算过程用该程序设计语言表示出来就是计算机程序。这个过程就是“程序设计”。

这就是一个简单的实际的计算问题,也是一个简单的程序设计过程。一般的,一个现实问题要使用计算机来解决,大致经过如下步骤:

- ① 建立数学模型——把实际问题转化为数学问题;
- ② 找出计算方法——为数学问题的求解找出方法;
- ③ 进行算法分析——为实现计算方法给出具体算法;
- ④ 选择一种程序设计语言,编出计算机程序——写程序;
- ⑤ 调试程序——保证程序的正确性;
- ⑥ 上计算机运行,测试程序的正确性——组装测试、确认测试;
- ⑦ 最后交付使用并维护。

在例 1.1 简单问题的解决过程中:

- 第一步分析就是建立数学模型,列出了二元一次方程组。
- 第二步分析为找出计算方法,选择“消去法”解二元一次方程组。
- 第三步分析是算法分析,得到了解二元一次方程组的具体计算步骤。
- 第四步写程序、调试程序、测试程序本例没有给出过程。
- 最后是交付使用并维护。

1.2 算 法

1.1 节中最后给出的计算步骤①~⑥是解决例 1.1 问题的算法。“程序设计”的任务就是找出算法(算法分析)、编出计算机程序、调试测试程序和运行程序。现实世界是五花八门、十分复杂的,要解决算法问题需要靠长期的学习、积累和悟性。选择一种语言,要根据具体问题来决定,本书以 C 语言为背景讲授“程序设计”,也就是说选择了 C 语言。程序设计是实践性很强的课程不要忽略上计算机进行调试,任何程序不经过调试是不能保证其正确性的。

算法由某些基本的成分组成,这些基本成分是一些基本的操作及控制结构。构成算法的基本操作包括:

- 表达式以及给变量赋值;
- 读(输入);
- 写(输出)。

基本的控制结构包括:

- 顺序控制结构;
- 分支控制结构;

- 循环控制结构；
- 函数调用和返回。

算法是一个计算过程,具体指明应该进行的操作,描述了解决问题的方法和途径,它是程序设计的基础和精髓。一个有效的算法具有如下特点:

1. 有穷性

一般情况下一个算法应该在有限的时间内终止,不应该是无限的。进一步,有穷性往往指在合理的时间范围内,比如后文讲述的 Hanoi 塔问题的循环迭代算法,若 1 秒钟计算一次,大约需要 5849 亿年。虽然是有穷的,但是显然是无意义的。

2. 确定性

算法中的每一个步骤都应该是确定的,含义是唯一的,不应该是模糊的、模棱两可的。例如求解一元二次方程,如果算法直接写成:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

显然是不确定的,不能构成确定的算法步骤。比如当 a 等于 0 时怎么办? 当 $b^2 - 4ac$ 小于 0 时怎么操作? 都没有确定的描述。

3. 有效性

算法中的每一个步骤都应该是有效的,都能够被有效的执行并得到确定的结果,不应该存在无效的操作。如在一个算法中存在操作“ $x \rightarrow x$ ”,显然无意义,也是无效的;又比如操作“ $a \div 0$ ”也是不能有效执行的。

1.3 PAD 图

描述算法有多种多样的方法。例如,流程图、NS 图、程序、PAD 图等。本书采用所谓的问题分析图(Problem Analysis Diagram, PAD)来描述算法,问题分析图简称 PAD 图。

PAD 图使用两维的树形结构描述程序的逻辑,因此它比直接用程序(可以说程序的表现形式是一维的)表示算法更清晰直观;PAD 图使用了结构化的概括的抽象的记号系统,所以它比用流程图表示算法更清晰、简练、紧凑、层次分明;PAD 是开放的,所以它比封闭式的 NS 图更清晰、分明,也更便于修改。

为简单明了起见及印刷上的原因,也为了适应 C 的一些特点,我们将标准 PAD 的记号系统作了一定的修正。这里先给出 PAD 记号系统的基本格式,再给出例 1.1 算法的 PAD 描述。更复杂控制结构的 PAD 表示将在下文涉及的地方逐步给出。

- PAD 把基本操作序列用方框括起来,表示成图 1.1 的形式。
- PAD 把顺序执行的操作用一条竖线顺序连接起来表示成图 1.2 的形式。竖线从上向下表示程序执行顺序,竖线连接起来的是一个操作成分。

【例 1.2】 例 1.1 问题算法(计算步骤①~⑥)的 PAD 描述。

解: PAD 描述如图 1.3 所示。



图 1.1 基本操作序列

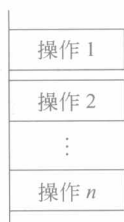


图 1.2 顺序控制结构

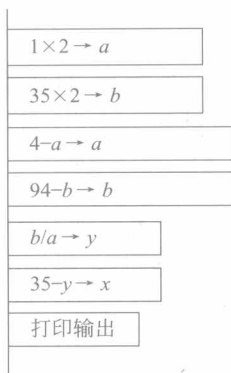


图 1.3 例 1.1 算法的 PAD 描述

1.4 程 序

一个庞大的计算机系统是怎样有条不紊的工作的呢？答案是：计算机系统的工作是由事先设计好的程序来控制的。人们首先按自己的需要，把让计算机做的工作编排成计算机程序，并把程序送入计算机，然后启动计算机执行程序。计算机的控制器从程序的第一条指令开始，顺序的逐条取出指令进行解释，然后按指令的规定和要求指挥整个计算机系统的工作，从而完成人们设想的要计算机完成的工作。

程序是一个指令序列，也就是用指令排成的一个工作顺序，工作步骤。我们平常也使用程序这个名词，例如运动会程序等。计算机程序是用计算机指令为计算机排定的工作顺序、工作步骤。为计算机编排程序的过程称为程序设计。

描述程序必须使用一种语言。程序设计语言是指用于编写、描述计算机程序的语言。C 语言是使用最广泛的程序设计语言，本书以 C 语言为载体讲述程序设计。

下面将例 1.1 的计算过程，按图 1.3 的 PAD，写出 C 程序。

【例 1.3】 例 1.1 的 C 程序。

解：程序代码如下。

```
#include <stdio.h>                // 插入标准输入输出函数库头文件
int a,b,x,y;                       // 声明 5 个变量, 分别表示计算用的中间结果, 和最后结果
void main(void) {                  // 主函数
    a=1*2;                          // 方程 (1) 乘以 2
    b=35*2;
    a=4-a;                          // 方程 (2) 减方程 (3)
    b=94-b;
    y=b/a;                          // 求 y
    x=35-y;                          // 求 x
    printf("鸡: %3d 兔: %3d\n",x,y); // 打印输出
}
```

这是一个完整的 C 程序，该程序全部在一个文件中。在该程序中，从