

21 世纪高等学校计算机教育实用规划教材

C程序设计实用教程

张述信 编著



清华大学出版社

21

世纪高等学校计算机教育实用规划教材

C程序设计实用教程

张述信 编著

图书在版编目(CIP)数据

C程序设计实用教程/张述信编著. —北京: 清华大学出版社, 2003.7

(21世纪高等学校计算机教育实用规划教材)

ISBN 958-7-305-19032-9

I. C… II. 张… III. C语言—程序设计—师范大学—教材 IV. TP312

中国版本图书馆CIP数据核字(2003)第028291号

责任编辑: 李海英

封面设计: 李海英

版式设计: 李海英

出版发行: 清华大学出版社

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

社 址: 北京 010-62770175

社 址: 北京 010-62770175

社 址: 北京 010-62770175

社 址: 北京 010-62770175

社 址: 北京 010-62770175

社 址: 北京 010-62770175

社 址: 北京 010-62770175

社 址: 北京 010-62770175

社 址: 北京 010-62770175

清华大学出版社
北京

本书如存在文字不清、漏印、错印、缺页、倒页、张数不符等质量问题, 请寄回本社出版部调换。联系电话: (010) 62770175 转 3103 产品编号: 03282-01

内 容 简 介

C 语言是当代计算机的主流语言，是学习程序设计的首选语言。

全书紧紧围绕着 C 程序设计这个主题展开，由四条干线贯穿始终。一是 C 语言的基本知识，包括基本概念、数据类型、运算符与表达式、控制台与文件的输入输出；二是 C 程序的基本结构，包括顺序结构、选择结构、循环结构和函数结构；三是 C 程序设计的基本技术、方法和技巧，如穷举、递推、递归、菜单制作与程序优化等；四是 C 程序上机操作，包括集成开发环境、程序测试与调试等。

通俗易懂地引入概念，循序渐进地讲解知识，深入浅出地化解难点，实用全面的设计技术，这是本书的主要特点。本书特别适合 C 程序设计的初学者使用，非常适用“以学生自学为主，教师重点讲解为辅”的方式教学。本书可作为高等院校计算机专业及理工科其他专业学习 C 程序设计的本科教材，亦可作为广大教师、工程技术人员与程序设计爱好者拓宽应用的参考书。

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

C 程序设计实用教程/张述信编著. —北京：清华大学出版社，2009.7

(21 世纪高等学校计算机教育实用规划教材)

ISBN 978-7-302-19935-9

I. C… II. 张… III. C 语言—程序设计—师范大学—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 058591 号

责任编辑：郑寅堃 徐跃进

责任校对：梁 毅

责任印制：孟凡玉

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 装 者：北京国马印刷厂

经 销：全国新华书店

开 本：185×260 印 张：21.25 字 数：528 千字

版 次：2009 年 7 月第 1 版 印 次：2009 年 7 月第 1 次印刷

印 数：1~4000

定 价：29.00 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：032665-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的进一步完善,社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,合理调整和配置教育资源,在改革和改造传统学科专业的基础上,加强工程型和应用型学科专业建设,积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业,积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新教学内容、改革课程体系,使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用,工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前,工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践,如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材),重理论、轻实践,不能满足新的教学计划、课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等,都不利于学生能力的提高和素质的培养。为此,在教育部相关教学指导委员会专家的指导和建议下,清华大学出版社组织出版本系列教材,以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

(2) 反映教学需要,促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有

本具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人: 丁岭 dingl@tup.tsinghua.edu.cn

前言

电子计算机俗称电脑。那么，学习电脑主要学什么？一要学操作，二要学程序设计。所谓操作电脑就是学会常用软件的使用，如 Windows、IE、Word、Excel 等，以便应用于工作、学习、交流和管理。仅此还远远不够，更重要的是学习如何用计算机语言设计程序，它是软件的核心。大家知道，世界个人电脑软件业霸主——美国微软公司，前董事长比尔·盖茨就是靠程序设计起家的，靠开发软件发达的，使他蝉联世界首富十多年。软件作为 20 世纪人类文明进步的最伟大成果之一，代表了先进文化的前进方向。美国政府早在 1992 年“国家关键技术”一文中就指出：“在很多国家关键技术中，软件是关键的、起推动作用（或阻碍作用）的因素”。从计算机应用的广度和深度上，其蓬勃发展的六十余年历史都充分证明了“计算机程序设计语言是第二文化”^①的科学论断。

学习程序设计的首选语言是诞生于 20 世纪 70 年代初的 C 语言，它是当代最有影响、最为流行的程序设计语言之一。由于 C 语言具有高级语言的结构，并提供类似汇编语言那样的系统资源操纵能力，使用它不但可以编写系统软件（如操作系统、数据库管理系统），而且可以编写应用软件。C 语言以其功能强大、执行效率高、简洁灵活、可移植性好深受广大程序设计者的青睐，成为当代计算机的主流语言。C 语言是面向过程的程序设计语言，它是进一步学习面向对象的程序设计语言（如 C++、VC++、C#、Java 等）的基础。因此，C 程序设计是高等学校计算机专业的主干课，理工科其他专业的必修课。

读者不禁要问：我能学会 C 程序设计吗？这里有 3 道自测题，如果你能独立解答，说明你具备了学习程序设计的能力。

- (1) 大小相同的两个玻璃杯中分别装有水和酒，问如何交换杯中的液体？
- (2) 一个地质小分队有 20 人，探矿途中遇到一条河无法过去。正在着急之时，恰有两少年划过来一条小船。这条船一次只能载一个队员或两个少年，问地质队员如何过河？
- (3) 如下图所示，A 处堆放 3 个木块，大块在下，小块在上。请你将 A 处木块搬到 C 处（可借助他处），要求一次只能移动一个木块，并且小块必须在大块上面。问如何实现？



你心里有底了吧。只要你接受过中等教育，具有计算机的基本知识，就能迈进程序设计的大门。遵循“认真看书、专心听讲、上机实习、独立做题”的学习路线，就能逐步掌握程序设计的真谛。坚持不懈地努力，必然获得丰硕的成果。

^① 1981 年在瑞士洛桑召开的第三次世界计算机教育大会上，前苏联学者伊而肖夫首次提出。

全书共 12 章,紧紧围绕着 C 程序设计这个主题展开,由四条干线贯穿始终:一是 C 语言的基本知识,包括基本概念、数据类型、运算符与表达式、控制台与文件的输入输出;二是 C 程序的基本结构,包括顺序结构、选择结构、循环结构和函数结构;三是 C 程序设计的基本技术、方法和技巧,如穷举、递推、递归、菜单制作与程序优化等;四是 C 程序上机操作,包括集成开发环境、程序测试与调试等。

为初学者编写的这本 C 教材具有几个较鲜明的特点:

(1) 从实际问题特别是日常实例引入重要概念。例如,从交换两个杯中的液体给出程序概念;由过河问题提出循环结构;通过如何输出高于平均分的成绩引进数组等。

(2) 深入浅出地化解知识难点。例如:

● 递归是实用而难掌握的技术。本书从老和尚讲故事谈起,用几个具体例子比较递归与循环的构造异同就不难理解了。

● 二维数组的地址是复杂的,通过降维即看成由几个一维数组组成的就简单了。

● 单向链表作为结构的重要应用,不但概念比较难懂,操作也难以把握。为此,之前添加一个动态数组分配与操作的例子,再理解单向链表就容易多了。

(3) 强调程序设计技术、方法和技巧,重视知识的综合运用。例如,判断一个自然数是否为质数,逐步给出优化的 3 种方法;起泡排序,除常规方法外又给出优化方法;求二维整型数组的每行最大值,共给出 9 种方法,将“指针”一章的核心内容与主要方法紧密联系起来。

(4) 注重知识的开发与应用。例如,使用 `const` 定义符号常量,并应用于共享数据的保护;尾递归调用;函数指针数组设计;实型数据的存储格式;以及面向过程的程序设计等。这些内容在同类教材中是难以寻觅的。

(5) 实用而丰富的操作实践。除每章专门开辟“上机指导”一节外,正文中不时指出上机遇到的问题和解决办法,并适当安排一些程序测试习题。

总之,这本教程内容翔实,面向应用,视野开阔,易教易学。本书特别适合 C 程序设计的初学者使用,非常适用“以学生自学为主,教师重点讲解为辅”的方式教学。本书可作为高等院校计算机专业及理工科其他专业学习 C 程序设计的本科教材,亦可作为广大教师、工程技术人员与程序设计爱好者拓宽应用的参考书。

囿于作者的能力和水平,书中难免存在错误或不当之处,请读者不吝赐教。

作者

2008 年 12 月



本书可作为高等院校计算机专业及理工科其他专业学习 C 程序设计的本科教材,亦可作为广大教师、工程技术人员与程序设计爱好者拓宽应用的参考书。

目 录

第1章 C 程序概述	1
1.1 什么是程序	1
1.2 什么是计算机程序	1
1.3 简单的 C 程序结构	3
1.4 上机指导 1	4
习题 1	6
第2章 C 语言基本知识	8
2.1 常量与变量	8
2.2 整型数据	9
2.2.1 整型常量的表示方法	9
2.2.2 整型变量的类型	9
2.2.3 整型常量的类型	11
2.3 实型数据	11
2.3.1 实型常量的表示方法	11
2.3.2 实型变量的类型	12
2.4 字符型数据	13
2.4.1 ASCII 标准字符	13
2.4.2 字符常量的表示方法	14
2.4.3 字符变量的类型	15
2.5 符号常量	16
2.6 算术运算符	16
2.6.1 基本算术运算符	17
2.6.2 负号运算符	18
2.6.3 自增与自减运算符	18
2.6.4 算术表达式	19
2.7 赋值运算符	20
2.7.1 基本赋值运算符	20
2.7.2 算术赋值运算符	20
2.7.3 赋值表达式	20
2.8 强制类型转换运算符	22
2.9 关系运算符	23
2.9.1 关系运算符	23
2.9.2 关系表达式	23

2.10 逻辑运算符	24
2.10.1 逻辑运算符	24
2.10.2 逻辑表达式	25
2.11 条件运算符	26
2.12 逗号运算符	28
2.13 长度运算符	29
2.14 数学函数	30
2.15 上机指导 2	31
习题 2	32
第 3 章 顺序结构程序设计	36
3.1 基本语句	36
3.2 格式化输入输出函数	37
3.2.1 格式化输入函数 scanf	37
3.2.2 格式化输出函数 printf	39
3.3 字符输入输出函数	42
3.3.1 字符输入函数 getchar	42
3.3.2 字符输出函数 putchar	43
3.4 上机指导 3	44
习题 3	45
第 4 章 选择结构程序设计	48
4.1 什么是选择结构	48
4.2 条件语句	48
4.2.1 基本格式	48
4.2.2 简化格式	49
4.3 条件语句的嵌套	51
4.3.1 条件语句嵌套示例	51
4.3.2 else if 格式	52
4.4 开关语句	54
4.5 上机指导 4	56
习题 4	58
第 5 章 循环结构程序设计	61
5.1 什么是循环结构	61
5.2 while 语句	61
5.3 do~while 语句	66
5.4 for 语句	68
5.5 循环的终止与继续	71
5.5.1 break 语句	71
5.5.2 continue 语句	74
5.6 循环语句的嵌套	75
5.7 用 goto 语句构造循环	78

5.8	结构化程序设计	79
5.9	上机指导 5	81
	习题 5	83
第 6 章	数组	89
6.1	数组的由来	89
6.2	一维数组	89
6.2.1	一维数组的定义与引用	89
6.2.2	一维数组的初始化	91
6.3	二维数组	95
6.3.1	二维数组的定义与引用	95
6.3.2	二维数组的初始化	98
6.4	字符串及其处理	100
6.4.1	字符串常量	100
6.4.2	字符数组	101
6.4.3	字符串的输入输出	104
6.4.4	字符串处理函数	107
6.5	上机指导 6	110
	习题 6	112
第 7 章	函数结构程序设计	118
7.1	什么是函数结构	118
7.2	函数定义	119
7.2.1	函数定义的一般形式	119
7.2.2	函数的返回值	120
7.3	函数调用	122
7.3.1	函数调用的一般形式	122
7.3.2	函数参数的传递	122
7.3.3	对被调函数的声明	124
7.3.4	函数的嵌套调用	126
7.4	函数的递归调用	127
7.4.1	尾递归	128
7.4.2	一般递归	129
7.5	数组名作函数参数	134
7.5.1	一维数组名作参数	134
7.5.2	二维数组名作参数	139
7.5.3	字符数组名作参数	140
7.6	变量的作用域和生存期	142
7.6.1	局部变量	142
7.6.2	全局变量	144
7.6.3	变量的存储属性	148
7.7	模块化程序设计	148
7.8	上机指导 7	152

习题 7	153
第 8 章 编译预处理	162
8.1 宏定义命令	162
8.1.1 简单的宏定义	162
8.1.2 带参的宏定义	163
8.2 文件包含命令	165
8.3 条件编译命令	166
8.4 上机指导 8	168
习题 8	169
第 9 章 指针	173
9.1 什么是指针	173
9.2 指向变量的指针变量	174
9.3 指向一维数组的指针变量	176
9.3.1 一维数组的地址	176
9.3.2 指针变量的定义和使用	178
9.3.3 指针变量的运算	179
9.3.4 指针变量可以带下标	181
9.3.5 指针变量作函数参数	182
9.3.6 指向字符串的指针变量	184
9.4 指针数组	187
9.4.1 指针数组的定义与使用	187
9.4.2 指向指针的指针变量	191
9.4.3 带参数的 main 函数	193
9.5 二维数组的指针变量	196
9.5.1 二维数组的地址	196
9.5.2 指向二维数组的指针变量	199
9.5.3 指向二维数组的指针变量作函数参数	200
9.6 指向函数的指针变量	202
9.7 返回指针的函数	204
9.8 动态内存分配	208
9.8.1 指针转换	208
9.8.2 void 型指针	209
9.8.3 动态内存分配	210
9.9 共享数据的保护	213
9.10 上机指导 9	214
习题 9	215
第 10 章 结构、联合与枚举	227
10.1 结构的含义	227
10.2 结构及其变量的定义	228
10.2.1 结构的一般形式	228

10.2.2	引用结构变量成员的方法	228
10.2.3	定义结构变量的其他方法	230
10.3	结构数组	231
10.4	指向结构数据的指针变量	232
10.4.1	指向结构变量的指针变量	232
10.4.2	指向结构数组的指针变量	233
10.5	向函数传递结构数据	234
10.6	结构成员的类型	237
10.7	单向链表	243
10.7.1	单向链表的概念	243
10.7.2	单向链表的建立	244
10.7.3	单向链表的其他操作	245
10.8	联合	249
10.9	枚举	250
10.10	数据类型重命名	252
10.11	上机指导 10	254
	习题 10	256
第 11 章	位运算	260
11.1	二进制数	260
11.1.1	无符号十进制整数转换成二进制整数	260
11.1.2	二进制整数转换成十进制整数	261
11.1.3	二进制与八、十六进制间的对应关系	262
11.2	位运算符	263
11.2.1	位与运算符	263
11.2.2	位或运算符	263
11.2.3	位异或运算符	264
11.2.4	位左移运算符	265
11.2.5	位右移运算符	265
11.2.6	位反运算符	267
11.2.7	位赋值运算符	267
11.3	位域	269
11.4	实型数据的存储格式	270
11.5	上机指导 11	273
	习题 11	274
第 12 章	文件	276
12.1	什么是文件	276
12.2	磁盘文件	276
12.2.1	文件的组织方式	276
12.2.2	文件的处理过程	277
12.3	磁盘文件的打开与关闭	278
12.3.1	文件的打开	278

12.3.2 文件的关闭	279
12.4 磁盘文件的输入输出	280
12.4.1 以字符为单位的读写	280
12.4.2 以字符串为单位的读写	284
12.4.3 以格式化数据为单位的读写	286
12.4.4 以数据块为单位的读写	288
12.5 磁盘文件的定位和随机处理	291
12.5.1 fseek 函数	292
12.5.2 ftell 函数	294
12.5.3 rewind 函数	294
12.6 磁盘文件的出错检测	296
12.6.1 ferror 函数	296
12.6.2 clearerr 函数	296
12.7 设备文件	297
12.7.1 设备文件的处理	297
12.7.2 标准设备文件的重定向	299
12.8 面向过程的程序设计	300
12.9 上机指导 12	302
习题 12	303
附录 A ASCII 标准字符集	309
附录 B C 运算符	310
附录 C C 语言关键字	311
附录 D C 常用库函数	312
D1 数学函数	312
D2 输入输出函数	312
D3 字符函数	314
D4 字符串函数	314
D5 动态存储分配函数	315
D6 其他函数	315
附录 E Turbo C 2.0 集成开发环境	316
E1 从 Windows 进入 DOS	316
E2 安装 TC	316
E3 启动 TC	317
E4 TC 菜单	318
E5 编辑窗口与常用编辑键	319
E6 信息窗口与快速参考行	320
参考文献	322

第1章

C 程序概述

1.1 什么是程序

例如，有这样一个问题：在大小相同的甲、乙两个玻璃杯中，分别装有 350 毫升水和 500 毫升酒（见图 1-1）。问如何交换两个杯中的液体？

这个问题不难，只要稍加思考就能回答。拿来一个和这两个杯子大小相同的空杯，称为丙杯。然后，按下面的操作即可实现：

- (1) 将甲杯中的水倒入丙杯；
- (2) 将乙杯中的酒倒入甲杯；
- (3) 将丙杯中的水倒入乙杯。

其过程如图 1-2 所示。把上面的操作步骤叫做解决“交换两个杯中液体”问题的程序。



图 1-1

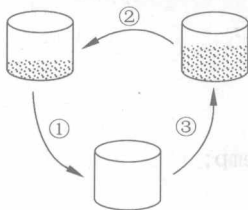


图 1-2

一般地，人们在日常生活、工作学习、生产劳动、科学实验等活动中，常常将完成某件事情的步骤集合称为程序。例如，民事诉讼法规定的开庭审理程序包括以下 6 个步骤：

- 开庭准备；
- 宣布开庭；
- 法庭调查；
- 法庭辩论；
- 合议庭评议；
- 宣告判决。

其他如学生伤害事故处理程序、国家助学贷款申请程序、财务收支审计程序，以及关于悬赏特异功能人的具体操作程序等。可见，程序这个术语是一个大众化的词语。

1.2 什么是计算机程序

前面提到的程序是用自然语言（如汉语言）来表达的。本书的重点是讲解计算机程序，指用计算机语言（例如 C 语言）来表示的解题步骤。那么，这样的程序是什么样子呢？

为此,首先要搞清楚解题所用的数据存放在哪儿?是计算机的内存储器,简称内存。内存是由一个个大小相同的“小房间”——字节(B)组成的,1个字节可存储一个西文字符(如A),2个连续的字节可存储一个整数(如35),4个连续的字节可存储一个实数(如2.5)。为了使用方便,将一个字节或几个连续的字节称为一个存储单元,并为其命名叫做变量名(借用数学上的术语)。如图1-3所示,名为ch、a和x的存储单元分别存放字符A、整数35和实数2.5。这些数据是存储单元的内容,或者说是变量的值。显而易见,存储单元与其内容(或者说变量与其值)类似于杯子和所装的液体。

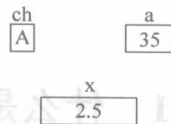


图 1-3

其次,要知道如何将数据存入变量所代表的存储单元中去,以及怎样对存储单元的内容进行操作。下面结合具体的例子来回答。

例 1.1 将整数 350 和 500 分别存入到两个存储单元中,然后交换它们的内容。

解答这个问题,从本质上讲和“交换两个杯中的液体”是一样的。区别在于,现在要用计算机语言——C 语言来表示操作步骤。将解题用的数据存入到存储单元的最简单方法是赋值,使用的格式如下:

变量名=数据

其中的等号(=)叫赋值运算符。

程序全文如下:

```
main( )
{
    int a,b,temp;           /*定义 3 个变量,代表 3 个存储单元*/
    a=350;                  /*将 350 存入单元 a 中*/
    b=500;                  /*将 500 存入单元 b 中*/
    temp=a;                 /*将 a 中内容存入 temp 单元中*/
    a=b;                    /*将 b 中内容存入 a 单元中*/
    b=temp;                 /*将 temp 中内容存入 b 单元中*/
    printf("a=%d,b=%d\n",a,b); /*输出 a、b 单元的内容*/
}
```

程序行中的/*...*/之间的部分是注释,用于说明程序的含义。注释不属于程序正文,目的在于增强程序的可读性。第3行定义3个存储单元,分别用变量名a、b、temp表示。其前的int表示变量的类型,指出每个变量在内存占2字节,单元的内容只能是整数。第4~第8行是赋值操作,其含义是将其右侧的数据(如350)或单元(如a)的内容存入到左侧变量所代表的存储单元中去。其中,第6~第8行的操作过程如图1-4所示。

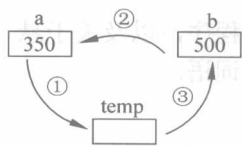


图 1-4

程序第9行是输出操作,双撇号(不是中文的双引号)内的%d是格式说明符,表示将变量a和b的值以十进制整数形式显示在屏幕上。双撇号中的“a=”与“,b=”是普通字符,原样显示,用来间隔输出数据。而\n是换行符,它使光标进到下一行的开头。程序执行结果是:

```
a=500, b=350
```

例 1.2 求任意两个整数的和并输出。

通过赋值操作可将数据存入到变量所代表的存储单元中。此外，C 语言还提供从键盘输入数据给变量，这种方式具有更大的灵活性。

```
main( )
{
    int a, b, sum;           /*定义 3 个变量，代表 3 个存储单元*/
    scanf("%d%d", &a, &b);   /*从键盘输入两个整数给变量 a 和 b*/
    sum=a+b;                 /*求 a、b 值的和，将结果赋值给变量 sum*/
    printf("Sum=%d\n", sum); /*输出变量 sum 的值*/
}
```

程序第 4 行是输入操作，双撇号内的 %d 指示从键盘输入十进制整数存入到其后的变量 a 和 b 所代表的存储单元中去，变量名前必须加上取地址运算符(&)。第 5 行是先对 a、b 的值求和，然后将结果赋值给变量 sum。第 6 行双撇号中的“Sum=”是普通字符序列，原样输出，即在对应位置上显示所写的字符。

程序运行时，从键盘输入任意两个整数。例如：

```
35 50↵
```

输入的数据一般用空格间隔，结束按 Enter 键（常叫回车键，用↵表示）。输出结果是：

```
Sum=85
```

由上面两个程序可知，对存储单元的内容进行操作是通过该单元的变量名和运算符来实现的。例 1.1 的“a=350;”表示将 350 存入 a 所代表的存储单元中。例 1.2 的“sum=a+b;”表示从 a 中取出 35，从 b 中取出 50，在运算器相加后，再将结果 85 存入 sum 所表示的存储单元中，其操作过程如图 1-5 所示。

与自然语言程序截然不同的是，计算机程序的最大特点是符号化：存储单元用符号表示，如 a、b、temp 等表示不同的存储单元；操作也用符号表示，如=表示赋值操作，+表示加运算。花括号（指{与}）内每个以英文分号结尾的符号序列叫做一个语句，类似于中文的句子。

可见，计算机程序是指“符号化的语句序列”^①，而编制程序的过程称为程序设计。

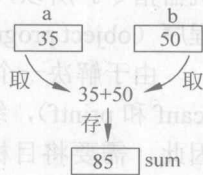


图 1-5

1.3 简单的 C 程序结构

上节的两个 C 程序，尽管它们的功能不同，但都称为 main 函数（借用数学上的术语），或叫做主函数。要注意，main 后面的圆括号不可省略，它是函数的标志。所谓简单的 C 程

① 计算机软件保护条例，中华人民共和国国务院令，该法于 1991 年 10 月 1 日起施行。

序,是指仅编写一个 main 函数的程序,其一般格式如下:

```
main()  
{ 定义变量  
  变量置值  
  加工处理  
  结果输出  
}
```

其中,第一个左花括号表示函数的开始,最后一个右花括号表示函数的结束,两个花括号之间的部分称为函数体。函数体内的第一步是定义变量,即为变量命名并指出变量的类型。第二步是为变量置值,常通过赋值或键盘输入。第三步是加工处理,就是对数据进行各种运算。第四步是结果输出,常把操作结果显示在屏幕上。第一步是程序的说明部分,后面 3 步是程序的执行部分。

强调指出,程序中的 scanf 和 printf 也叫函数,它们是由系统提供的现成函数,分别用来完成数据的输入操作和输出操作。学会使用这两个函数并不复杂,逐渐就熟悉了。由此可见,只编写一个 main 函数的 C 程序,其核心是加工处理步骤。

用计算机解决各种各样的问题,就要编写像例 1.1 和例 1.2 这样的程序,即解题步骤由符号化的语句组成,我们把它叫做源程序(source program)。上机对源程序进行操作之前,需要明确几个重要概念。

首先要将源程序通过键盘(一般是这样)输入到计算机中并存盘。如果发现有文字错误(如 a 写成 A)、语法错误(如 35=a)或流程错误(如算法不对)等,需要修改源程序,之后再存盘,这样的过程叫做编辑。

计算机是不“认识”源程序语句的,它只能识别由 0 和 1 组成的机器指令。所以,要把源程序翻译成机器指令组成的程序,称为目标程序(object program),这一步叫做编译。

由于解决一个问题的 C 程序中经常使用系统提供的现成函数(如 scanf 和 printf),统称为库函数,它们实质上是完成某项功能的程序。因此,需要将目标程序和所使用的库函数装配成一个可执行的程序(execution program),这个过程叫做连接。

运行这个可执行程序,输入程序所要求的数据,通过显示或打印得到所需要的结果,这一步叫做运行。

上机操作一个 C 程序,一般要经过 4 大步骤:编辑、编译、连接和运行。不管哪一步,都可能出现错误。因此,修改程序错误在所难免,再重复这几步,直到获得正确的结果。图 1-6 中的 c、obj 和 exe 分别是源程序文件、目标程序文件和可执行程序文件的扩展名。

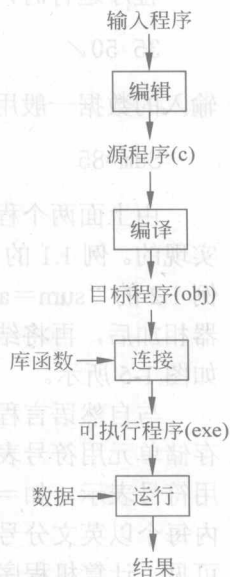


图 1-6

1.4 上机指导 1

上机操作一个 C 程序,通常使用 Turbo C 2.0 系统(简称 TC)提供的集成开发环境