



高等学校计算机教材建设立项项目

高 等 学 校 计 算 机 课 程 规 划 教 材

C 程序设计

钟家民 李爱玲 主 编
张 涵 姬秀嘉 张珊靓 王 璐 副主编



清华大学出版社



高等学校计算机教材建设立项项目

高等学校计算机课程规划教材

C程序设计

钟家民 李爱玲 主 编
张 涵 姬秀荔 张珊靓 王 璐 副主编

清华大学出版社
北京

内 容 简 介

本书对知识点的讲解由浅入深,强调算法设计,突出编程思路,注重实例讲解和对学生动手能力的培养。

全书共分10章,内容主要包括:C语言概述,C语言基础,选择结构程序设计,循环结构程序设计,数组,函数,指针,结构体与共用体,文件,综合实例程序设计。

本书适合作为高校计算机程序设计基础教材,也适合作为社会各界人士的自学参考书。

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/钟家民,李爱玲主编. —北京:清华大学出版社,2016

(高等学校计算机课程规划教材)

ISBN 978-7-302-43788-8

I. ①C… II. ①钟… ②李… 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100590号

责任编辑:汪汉友 柴文强

封面设计:傅瑞学

责任校对:李建庄

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:25.25

字 数:632千字

版 次:2016年8月第1版

印 次:2016年8月第1次印刷

印 数:1~2000

定 价:49.50元

产品编号:067305-01

出版说明

信息时代早已显现其诱人魅力,当前几乎每个人随身都携有多个媒体、信息和通信设备,享受其带来的快乐和便宜。

我国高等教育早已进入大众化教育时代。而且计算机技术发展很快,知识更新速度也在快速增长,社会对计算机专业学生的专业能力要求也在不断翻新。这就使得我国目前的计算机教育面临严峻挑战。我们必须更新教育观念——弱化知识培养目的,强化对学生兴趣的培养,加强培养学生理论学习、快速学习的能力,强调培养学生的实践能力、动手能力、研究能力和创新能力。

教育观念的更新,必然伴随教材的更新。一流的计算机人才需要一流的名师指导,而一流的名师需要精品教材的辅助,而精品教材也将有助于催生更多一流名师。名师们在长期的一线教学改革实践中,总结出了一整套面向学生的独特的教法、经验、教学内容等。本套丛书的目的就是推广他们的经验,并促使广大教育工作者更新教育观念。

在教育部相关教学指导委员会专家的帮助和指导下,在各大学计算机院系领导的协助下,清华大学出版社规划并出版了本系列教材,以满足计算机课程群建设和课程教学的需要,并将各重点大学的优势专业学科的教育优势充分发挥出来。

本系列教材行文注重趣味性,立足课程改革和教材创新,广纳全国高校计算机优秀一线专业名师参与,从中精选出佳作予以出版。

本系列教材具有以下特点。

1. 有的放矢

针对计算机专业学生并站在计算机课程群建设、技术市场需求、创新人才培养的高度,规划相关课程群内各门课程的教学关系,以达到教学内容互相衔接、补充、相互贯穿和相互促进的目的。各门课程功能定位明确,并去掉课程中相互重复的部分,使学生既能够掌握这些课程的实质部分,又能节约一些课时,为开设社会需求的新技术课程准备条件。

2. 内容趣味性强

按照教学需求组织教学材料,注重教学内容的趣味性,在培养学习观念、学习兴趣的同时,注重创新教育,加强“创新思维”,“创新能力”的培养、训练;强调实践,案例选题注重实际和兴趣度,大部分课程各模块的内容分为基本、加深和拓宽内容3个层次。

3. 名师精品多

广罗名师参与,对于名师精品,予以重点扶持,教辅、教参、教案、PPT、实验大纲和实验指导等配套齐全,资源丰富。同一门课程,不同名师分出多个版本,方便选用。

4. 一线教师亲力

专家咨询指导,一线教师亲力;内容组织以教学需求为线索;注重理论知识学习,注重学习能力培养,强调案例分析,注重工程技术能力锻炼。

经济要发展,国力要增强,教育必须先行。教育要靠教师和教材,因此建立一支高水平的教材编写队伍是社会发展的关键,特希望有志于教材建设的教师能够加入到本团队。通过本系列教材的辐射,培养一批热心为读者奉献的编写教师团队。

清华大学出版社

前 言

“C 程序设计”是计算机专业和非计算机专业的一门基础课程。通过本课程的学习,可以使学生更好地了解和应用计算机,培养学生应用计算机独立解决问题的能力,为今后进一步的学习奠定良好的、扎实的计算机语言基础。

本书是学习 C 语言程序设计的基础教材,由教学经验丰富的一线教师精心组织了教材的内容。对 C 语言的精华部分作了较为详细的介绍;较难的题目给出编程思路;还针对学生学完 C 语言后普遍感觉提高和综合应用难的问题,在最后一章安排了综合实例设计与分析,方便学生对全书内容的综合理解和应用;考虑到 C 语言程序设计是一门实践性比较强的课程,本教材在最后给出参考答案。

在教材的第 1、2 章,介绍了 C 语言的基本概念、各种数据类型;第 3、4 章介绍了 C 语言的基本程序设计技术;第 5、6 章介绍数组及函数的相互调用及变量的特性;第 7 章详细地介绍了指针的特点和灵活性;第 8、9 章介绍了结构体、链表技术和文件的操作方法;第 10 章是综合实例设计。附录给出了算法知识、常见编译错误信息、常用库函数等。列举的例题都是作者的精心设计,并全部在 Visual C++ 6.0 环境下调试通过。

本书的讲述深入浅出,配合典型例题,通俗易懂,实用性强,可作为高等院校计算机专业和非计算机专业本、专科学生的 C 语言教材,也可以作为自学者的参考用书。可免费给读者提供由本书作者开发的 C/C++ for Windows 程序设计与学习系统、书中源码和课件。

本书的第 1、2 章由张涵编写,第 3、5 由王璐编写,第 4、6 章钟家民编写并统稿,第 8、9 章由张珊靓编写,第 7、10 章由李爱玲编写,姬秀荔编写附录 A、附录 B、附录 C 及参考答案。

由于作者水平有限,书中难免会有不足和错误,希望读者和专家提出宝贵意见,以帮助我们将此教材进一步完善。

编 者

2016 年 6 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言引例	1
1.1.1 C 程序基本结构	3
1.1.2 C 程序书写格式	4
1.2 基本的输入输出函数的用法	4
1.2.1 scanf() 函数	4
1.2.2 printf() 函数	9
1.3 C 程序运行环境	12
1.3.1 Visual C++	12
1.3.2 Turbo C 下运行 C 语言源程序	16
1.4 算法	18
1.4.1 程序设计与算法	19
1.4.2 算法的描述	21
1.5 本章常见错误总结	24
本章小结	26
习题一	26
实验一	27
第 2 章 C 语言基础	29
2.1 基本的数据类型	29
2.1.1 标识符	30
2.1.2 关键字	30
2.2 常量	31
2.2.1 整型常量	31
2.2.2 实型常量	32
2.2.3 字符型常量	33
2.2.4 转义字符	33
2.2.5 符号常量	34
2.3 变量	34
2.3.1 整型变量	35
2.3.2 实型变量	37
2.3.3 字符型变量	39

2.3.4	sizeof()运算符	41
2.4	运算符和表达式	42
2.4.1	赋值运算符和赋值表达式	42
2.4.2	算术运算符	45
2.4.3	自增自减运算符	47
2.4.4	关系运算	48
2.4.5	逻辑运算	49
2.4.6	条件运算	51
2.4.7	位运算	52
2.4.8	逗号运算符和逗号表达式	55
2.5	顺序结构程序设计	56
2.5.1	C 语句	56
2.5.2	顺序结构程序举例	58
2.6	本章常见错误总结	59
	本章小结	61
	习题二	61
	实验二	66
第3章	选择结构程序设计	68
3.1	if 语句引例	68
3.2	if 语句	69
3.3	if 语句的嵌套	73
3.4	switch 语句	76
3.5	选择结构程序实例	81
3.6	常见错误	85
	本章小结	87
	习题三	87
	实验三	92
第4章	循环结构程序设计	96
4.1	循环引例	96
4.2	while 语句	97
4.3	do...while 语句	99
4.4	for 语句	101
4.5	break 和 continue 语句	104
4.5.1	break 语句	104
4.5.2	continue 语句	105
4.6	循环的嵌套	107
4.7	循环程序举例	109

4.8 本章常见错误总结	113
本章小结	115
习题四	116
实验四	119
第5章 数组	123
5.1 数组引例	123
5.2 一维数组	124
5.2.1 一维数组定义	125
5.2.2 一维数组引用和初始化	125
5.2.3 一维数组的应用	127
5.3 二维数组	132
5.3.1 二维数组的定义	132
5.3.2 二维数组的引用和初始化	133
5.3.3 二维数组程序举例	135
5.4 字符数组和字符串	138
5.4.1 字符数组	138
5.4.2 字符串	138
5.4.3 字符数组的输入输出方式	139
5.4.4 字符串处理函数	141
5.4.5 字符数组和字符串程序实例	145
5.5 数组实例	147
5.6 常见错误	151
本章小结	152
习题五	153
实验五	155
第6章 函数	160
6.1 函数引例	160
6.2 函数的定义与调用	162
6.2.1 函数的定义	162
6.2.2 函数调用	163
6.2.3 形式参数和实际参数	166
6.2.4 函数的返回值	167
6.3 函数的嵌套和递归	168
6.3.1 函数的嵌套	168
6.3.2 函数的递归调用	169
6.4 数组作为函数参数	173
6.5 变量的作用域及存储类型	178

6.5.1 静态、动态	179
6.5.2 变量的作用域	180
6.6 外部、内部函数	183
6.7 预处理命令	184
6.8 应用举例	186
6.9 本章常见错误总结	191
本章小结	193
习题六	194
实验六	197
 第7章 指针	200
7.1 指针引例	200
7.2 指针变量的定义和引用	200
7.2.1 指针变量的定义	201
7.2.2 指针变量的引用	202
7.2.3 指针变量作函数的参数	205
7.3 指针与数组	206
7.3.1 指针与一维数组	206
7.3.2 指针与二维数组	210
7.3.3 数组指针作函数的参数	212
7.3.4 指针与字符数组	217
7.3.5 指针数组	220
7.4 指针与函数	221
7.4.1 指向函数的指针变量的定义及使用	221
7.4.2 用指针类型数据作函数参数	223
7.4.3 带参的主函数	225
7.4.4 返回指针的函数	226
7.5 指向指针的指针	228
7.6 指针应用举例	229
7.7 指针常见错误小结	232
本章小结	234
习题七	235
实验七	237
 第8章 结构体与共用体	240
8.1 结构体引例	240
8.2 结构体类型声明与结构体变量定义	241
8.2.1 结构体类型声明	241
8.2.2 结构体类型定义	242

8.2.3	结构体变量的引用和初始化	244
8.2.4	结构体变量作为函数参数	247
8.3	结构体数组	249
8.3.1	结构体数组的定义	250
8.3.2	结构体数组的初始化	251
8.3.3	结构体数组作为函数参数	252
8.4	结构体指针	254
8.4.1	结构体指针变量的定义及引用	254
8.4.2	结构体数组指针	255
8.4.3	指向结构体的指针作为函数参数	256
8.5	链表——结构体应用	257
8.5.1	链表概述	258
8.5.2	链表基本运算	258
8.5.3	链表应用举例	261
8.6	共用体	264
8.6.1	共用体的定义	266
8.6.2	共用体的引用和初始化	267
8.7	枚举类型	268
8.7.1	枚举类型的声明和变量定义	269
8.7.2	枚举类型变量的操作	270
8.8	本章常见错误总结	271
	本章小结	274
	习题八	274
	实验八	276
第9章	文件	283
9.1	文件引例	283
9.2	文件概述	284
9.3	文件打开与关闭	284
9.3.1	文件的打开	284
9.3.2	文件的关闭	285
9.4	文件的读写	286
9.4.1	文件的字符读写	286
9.4.2	文件的字符串读写	290
9.4.3	文件的格式化读写	292
9.4.4	文件的数据块读写	293
9.5	其他文件函数	295
9.5.1	文件定位	295
9.5.2	文件检测	298

9.6 综合应用举例	300
9.7 本章常见错误总结	307
本章小结	308
习题九	308
实验九	310
第 10 章 综合实例程序设计	314
10.1 程序设计的基本过程	314
10.2 综合程序设计实例	315
10.2.1 题目的内容要求	315
10.2.2 程序的功能设计	316
10.2.3 程序的数据设计	317
10.2.4 程序的函数设计	318
10.2.5 函数编程及调试	319
10.2.6 整体调试	338
10.2.7 程序维护	338
10.3 C 语言大型程序项目的管理	338
本章小结	339
习题十	339
附录 A 常用 ASCII 码字符对照表	340
附录 B 编译错误信息	341
附录 C 常用库函数	354
附录 D 部分习题参考答案	363
习题一参考答案	363
习题二参考答案	364
习题三参考答案	366
习题四参考答案	368
习题五参考答案	371
习题六参考答案	373
习题七参考答案	378
习题八参考答案	383
习题九参考答案	386
参考文献	392

第 1 章 C 语言概述

随着时代的进步,计算机已经成为人们生活中不可或缺的一部分,但是你知道你使用的这些计算机软件是怎么来的吗?是程序设计人员编写出来的。

C 语言是一种计算机程序设计语言,属高级语言范畴。它既具有高级语言的特点,又具有汇编语言的特点。本章主要介绍 C 语言的特点,结构,基本的输入输出,C 语言的编译环境和算法。

1.1 C 语言引例

【例 1-1】 显示“This is a C program!”的 C 语言程序。

```
#include<stdio.h>                /* include 称为文件包含命令 */
void main()                       /* 函数首部 */
{
    printf("This is a C program!\n");    /* 通过 printf 函数,输出字符串 */
}
```

运行结果:

This is a C program!

本例很短,但五脏俱全。C 语言的源程序也叫源代码,字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空白符、标点和特殊字符组成。字母主要是以英语单词为主,如 include、void、main 等;符号不能随意输入,每种符号在 C 语言中都有特定的意义,后续章节会陆续说明。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。空白符主要有空格、制表符和换行符。

注意: 空白符只在字符常量和字符串常量中起作用。在其他地方出现时,只起间隔作用,编译程序对它们忽略不计。因此在程序中是否使用空白符,不会对程序的编译产生影响,但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

main 是 C 语言源程序中主函数的名字,void 是“空”的意思,用来说明 main()函数的类型,也就是指执行完 main 函数后无须产生函数值。

printf()是 C 语言函数库提供的输出函数,其功能是将小括号中的双引号内的字符原样输出。

【例 1-2】 从键盘输入两个整数,输出两个整数之和。

```
#include<stdio.h>                /* include 称为文件包含命令 扩展名为.h 的文件称为头文件 */
void main()                       /* 函数首部 */
{
    int a,b,sum;                  /* 定义三个整数类型变量 a,b,sum,以被后面程序使用 */
}
```

```

scanf("%d,%d",&a,&b);          /* 从键盘获得两个整数 a,b */
sum=a+b;                       /* 计算 a+b 的和,赋值给 sum */
printf("a 加 b 的和是%d ",sum); /* 通过 printf 函数,输出 sum 的值 */
}

```

运行结果:

```

9,10 ↵
a 加 b 的和是 19

```

以#开头的命令被称为预处理命令,预处理命令还有其他几种(详见后续章节)。include 称为文件包含命令,其意义是把尖括号<>或双引号"内指定的文件包含到本程序来,成为本程序的一部分。此处的 stdio.h 是系统提供的一个函数库文件名,全称是 standard input & output,即标准输入输出函数库(详见附录),凡是在程序中调用一个库函数时,都必须包含该函数原型所在的头文件。C 语言规定对 scanf() 和 printf() 这两个函数可以忽略对其头文件的包含预处理命令。

C 语言程序是由函数组成的。一个完整的函数应该包含两部分,即函数首部和函数体。在本例第二行是函数的首部,由一对大括号{}括起的内容称作函数体,函数体主要用来说明当前函数的功能如何实现;函数体又分为两部分,一部分为说明部分,另一部分为执行部分。说明部分指的是对变量的定义(函数类型和参数类型的说明)。例 1-1 中未使用变量,因此无说明部分。C 语言规定,源程序中所有用到的变量都必须先定义,后使用,否则将会出错。函数体内主要由语句来组成,语句最后必须有一个分号表示结束,即分号也属于 C 语句的一部分。

【例 1-3】 已知半径为 2.5cm,求圆面积。

```

#include<stdio.h>
void main()          /* 主函数 */
{
    float s (float x);
    float r,area;     /* 定义变量 r 和 area,并给 r 赋初值 2.5 */
    printf("请输入 r 的值\n");
    scanf("%f",&r);
    area=s(r);        /* 使用 s_area 函数,并将 r 的值传递给 s_area 函数的参数 */
    printf("圆面积为%f\n",area);
}
float s (float x)     /* 用来求圆面积的函数 s_area */
{
    return (3.14*x*x); /* 函数的返回值 */
}

```

运行结果:

```

圆面积为 19.625000

```

例 1-3 的功能是由用户输入 1 个小数给变量 r,程序执行后输出以 r 为半径的圆面积。本程序由两个函数组成,主函数和 s 函数。函数之间是并列关系。在给 area 变量赋值时,

程序从主函数中调用 s() 函数, 并将 r 的值传递给了 s() 函数中的变量 x。s() 函数的功能是计算圆面积, 将结果返回给主函数并且赋值给主函数中的 area 变量。s() 函数是一个用户自定义函数, 需要在主函数中给出说明(见程序第 3 行)。在程序的每行后用 /* 和 */ 括起来的内容为注释部分, 程序执行时不执行注释部分。注释的目的是让读者更方便地阅读和理解程序。

上述三个例题中的共同点是每个源程序都只有一个 main() 函数。

1.1.1 C 程序基本结构

C 语言属于结构化和模块化的程序设计语言, C 语言程序以函数作为程序的基本模块单位, 并具有结构化的控制语句(共 9 种)。

1. 程序的组成

每个程序一般由函数、编译预处理和注释 3 部分组成。

(1) 函数: 函数定义是 C 程序的主体部分, 程序的功能由函数来完成。

(2) 编译预处理: 每个以符号 # 开头的行, 称预处理, 是 C 提供的一种模块工具。

(3) 注释: 用 /* */ 括起的内容, 可以占用多行。其作用是给程序设计者一种提示或记号。注释内容不参加程序的执行, 主要是为了提高程序的可读性。

2. 函数的组成

(1) 每个函数(包括主函数)的定义分为两个部分: 函数首部和函数体。

(2) 函数首部包括: 函数类型、函数名和形式参数表。

函数首部的格式:

函数类型 函数名 (参数类型 参数名 1, 参数类型 参数 2, ..., 参数类型 参数 n)

例如:

```
float s (float x)
```

(3) 函数体是由一对 {} 括起的语句序列, 其中包括变量说明部分和实现函数功能的语句。

函数体的格式:

```
{  
    变量类型 变量名;  
    执行语句;  
    ...  
    输出语句;  
}
```

3. 程序结构特点

(1) 程序由一个或多个函数构成。每个 C 程序有且仅有一个主函数, 函数名规定为 main, 它由系统指定的。除主函数外, 可以有一个或多个子函数(如 s() 函数)。

(2) 主函数是整个程序的主控模块, 是程序的入口, C 程序运行时从 main() 函数开始执行, 最终在 main() 函数中结束。

(3) 函数在执行过程中可以根据程序的需要调用系统提供的库函数, 例如在程序用到

的 `scanf()` 或 `printf()` 函数, 它们的原型包含在 `stdio.h` 文件中。

(4) 源程序中可以有预处理命令(`include` 命令仅为其中的一种), 预处理命令通常应放在源程序的最前面。

(5) 每一个说明, 每一个语句都必须以分号结尾。

(6) 类型和函数名或变量名之间必须至少加一个空格间隔。

1.1.2 C 程序书写格式

C 语言语法限制不严, 程序设计自由度大, 但应书写清晰, 便于阅读、理解和维护, 在书写程序时应尽量遵循以下原则:

(1) 一个说明或一条语句占一行。`main()` 函数可以放在程序的任意位置。

(2) 大括号 `{}` 里的内容, 通常表示程序中的函数体或由多条语句所构成的复合语句结构。`{}` 一般与该结构语句的第一个字母对齐, 并单独占一行。

(3) 为增加程序的可读性。低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写, 以便看起来更加清晰。

1.2 基本的输入输出函数的用法

在运行程序时, 用户往往需要输入一些数据, 程序会根据输入的数据进行运算并将正确的结果输出给用户, 从而实现人与计算机之间的交互, 所以在程序设计中, 输入和输出是必不可少的操作。在 C 语言中, 没有专门的输入输出语句, 所谓输入输出是以计算机为主体而言的。所有的输入输出操作都需要借助于输入输出函数来实现, 因此都是函数调用语句。

这些函数的原型都写在 `stdio.h` 文件中, 故在程序中如果要使用输入输出函数, 则需要在程序开始部分进行说明。本节仅介绍格式化输入函数 `scanf` 和格式化输出函数 `printf()`, `printf()` 函数是向标准输出设备显示器输出数据的函数调用语句。在使用 C 语言库函数时, 要用预编译命令 `#include` 将有关“头文件”包括到源文件中。使用标准输入输出库函数时要用到 `stdio.h` 文件, 编译命令 `#include <stdio.h>` 或 `#include "stdio.h"`。

`stdio` 是 `standard input & output` 的意思。考虑到 `printf()` 和 `scanf()` 函数使用频繁, 系统允许在使用这两个函数时可不加 `#include <stdio.h>` 或 `#include "stdio.h"`。

1.2.1 scanf() 函数

`scanf()` 函数称为格式输入函数, 即按用户指定的格式从键盘上把数据输入到指定的变量之中, `scanf()` 函数是一个标准库函数。

`scanf()` 函数的一般形式为:

```
scanf("格式控制字符串", 地址表列);
```

其中, 格式控制字符串的作用与 `printf()` 函数相同, 程序设计人员可以在此处规定输入的具体格式。地址表列中列出接收数据的变量的地址。变量的地址是在变量名前加求地址运算符“&”。在 C 语言中, 计算机中的数据都要通过内存, 而内存的基本存储单位是字节, 也可以叫做存储单元, 每个存储单元都由系统分配一个编号, 这个编号就是地址。

例如：

`&a, &b`

`&` 是一个取地址运算符, `&a` 是一个表达式, 其功能是求变量的地址。分别表示变量 `a` 和变量 `b` 的地址。这个地址就是系统在内存中给 `a, b` 变量分配的地址。

变量的地址和变量值的关系如下：

在赋值表达式中给变量赋值, `a=56`

则, `a` 为变量名, 56 是变量的值, `&a` 是变量 `a` 的地址, 如图 1-1 所示。

但赋值号左边是变量名, 不能写地址, 可以用于在设计程序就给定的固定值, 而 `scanf()` 函数在本质上也是给变量赋值, 但要求写变量的地址, 如 `&a`, 程序运行后, 需要用户输入数值给变量。这两者在形式上和含义上都是不同的。

当输入多个数值数据时, 若格式控制串中没有非格式字符作输入数据之间的间隔则可用空格, Tab 或回车作间隔, C 语言程序编译时在遇到空格、Tab、回车或非法数据(如对“%d”, 输入“12A”时, 在 A 即为非法数据)时即认为该数据输入结束。

【例 1-4】 `scanf` 函数中连续输入数值型数据。

```
#include<stdio.h>
void main()
{
    int a, b, c;
    printf("input a, b, c\n");
    scanf("%d%d%d", &a, &b, &c);
    printf("a=%d, b=%d, c=%d", a, b, c);
}
```

运行结果：

```
input a, b, c
7 8 9 ↵
a=7, b=8, c=9
```

在例 1-4 中, 先用 `printf()` 语句在屏幕上输出提示, “输入 `a, b, c` 的值”。执行 `scanf()` 函数, 进入用户屏幕等待用户输入。用户输入 7 8 9 后按回车键。在 `scanf` 语句的格式串中, 由于没有普通字符在 `%d%d%d` 之间作输入时的间隔, 在输入时要用一个以上的空格、回车键或制表键 Tab 键作为每两个输入数之间的间隔。

输入字符数据时, 若格式控制串中无非格式字符, 则认为所有输入的字符均为有效字符。

例如：

```
scanf("%c%c%c", &a, &b, &c);
```

输入为：

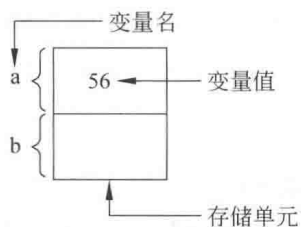


图 1-1 变量值与变量