全国高等职业教育规划教材

C语言程序设计教程

第3版

吉顺如　辜碧容　唐　政　编著

- 内容精炼，结构合理，重点突出，例题丰富。
- 概念清晰，逻辑性强，通俗易懂，实用性强。
- 实验内容，形式多样，循序渐进，提高实践编程能力。
- 习题丰富，题型多样，引导求知，培养良好编程风格。

机械工业出版社
CHINA MACHINE PRESS

全国高等职业教育规划教材

C语言程序设计教程

第3版

吉顺如　辜碧容　唐　政　编著

计春雷　主审

机械工业出版社

本书根据高校非计算机类专业“C 语言程序设计”课程教学大纲编写。在编写中仔细考虑了内容的取舍，突出对基本概念的讲解和叙述，将基本概念和方法的应用放在例题中，结合程序进行讲解，通俗易懂。本书共 9 章，内容包括 C 语言概述，数据类型、运算符和表达式，C 程序中的输入、输出，C 程序的控制结构，数组，函数，指针，结构体与共用体，文件等。每章精心选择典型例题进行分析，选择难易适中的习题供学生课后练习，每章的上机实验题均包括改错题、程序填空题及编程题。

本书适用于高职高专非计算机类专业的学生，也可供对程序设计有兴趣的读者参考。

本书配有授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（QQ：1239258369，电话：010-88379739）。

图书在版编目(CIP)数据

C 语言程序设计教程/吉顺如，辜碧容，唐政编著. —3 版. —北京：机械工业出版社，2015.3
全国高等职业教育规划教材
ISBN 978-7-111-49786-8

Ⅰ.①C… Ⅱ.①吉… ②辜… ③唐… Ⅲ.①C 语言—程序设计—高等职业教育—教材 Ⅳ.①TP312

中国版本图书馆 CIP 数据核字（2015）第 061263 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
责任编辑：王 颖 责任校对：张艳霞
责任印制：刘 岚
北京圣夫亚美印刷有限公司印刷
2015 年 5 月第 3 版 · 第 1 次印刷
184mm×260mm · 16 印张 · 392 千字
0001—3000 册
标准书号：ISBN 978-7-111-49786-8
定价：36.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
服务咨询热线：(010) 88379833	机 工 官 网：www.cmpbook.com
读者购书热线：(010) 88379649	机 工 官 博：weibo.com/cmp1952
	教育服务网：www.cmpedu.com
封面无防伪标均为盗版	金 书 网：www.golden-book.com

前　言

在 20 世纪 70 年代，C 语言就因为其高效性、灵活性和适应性而广为应用，迅速成为软件开发最主要的程序设计语言之一。随着计算机技术的飞速发展，虽然 C 语言在软件开发领域中的地位已逐渐为可视化编程语言（如 Visual Basic、Visual C++、Delphi 等）所替代，但是在工程应用领域，C 语言依然有着强大的生命力。特别在教育领域，C 语言仍是程序设计课程首选的入门语言。本书就是依据高职高专院校非计算机专业“C 语言程序设计”课程教学大纲编写的专用教材。通过本门课程的学习，使高职高专学生掌握 C 语言程序设计的基础知识、基本概念，掌握 C 语言程序设计的思想和编程技巧，通过实践，提高分析问题和解决问题的能力，为后续课程的学习和应用开发打下扎实的高级语言理论和实践基础。

本书在编写中仔细考虑了内容的取舍，以教学大纲为依据，不刻意追求“系统性和完整性”，而是把应用性作为重点。在教学内容的叙述上，突出基本概念，将基本概念和方法的应用放在例题中，结合程序进行讲解。同时，借助“程序说明”和“注意”等教学提示，帮助学生理解教学内容，少走弯路。为了帮助学生掌握有关的基本概念和方法，每章都精心选择了典型例题进行分析，选择难易适中的习题供学生课后练习。C 语言程序设计是一门理论性、实践性均较强的课程，要注重上机编程实践，因此本书的每个章节后均提供上机实验题，题型包括改错题、程序填空题及编程题。这些练习和实验编程的内容紧扣大纲要求，既有基本练习题，也配有少量有一定难度的题目，教师可根据实际教学情况选用。

本书由吉顺如、辜碧容、唐政编写，吉顺如统稿。全书的例题和习题均上机进行了调试验证。

限于编著者的学识水平，且由于时间仓促，书中错误在所难免，恳请读者提出宝贵意见。

编　者

目　录

前言
第 1 章　C 语言概述 …… 1
1.1　C 语言简介 …… 1
1.1.1　C 语言的产生 …… 1
1.1.2　C 语言的特点 …… 1
1.2　C 程序的结构及书写格式 …… 2
1.2.1　C 程序的结构 …… 2
1.2.2　C 程序的书写格式 …… 4
1.3　C 程序的开发过程 …… 4
1.4　典型例题分析 …… 5
1.5　实验 1　C 程序运行环境及简单程序的运行 …… 7
1.6　习题 …… 9
第 2 章　数据类型、运算符和表达式 …… 12
2.1　概述 …… 12
2.2　常量 …… 12
2.3　变量 …… 14
2.3.1　变量的概念 …… 14
2.3.2　变量的类型 …… 14
2.3.3　变量的定义和初始化 …… 15
2.3.4　各类数值型数据间的混合运算 …… 16
2.4　算术运算符和算术运算表达式 …… 16
2.4.1　算术运算符 …… 16
2.4.2　算术运算表达式 …… 18
2.5　赋值运算符和赋值表达式 …… 18
2.5.1　赋值运算符和复合的赋值运算符 …… 18
2.5.2　赋值运算表达式 …… 19
2.6　自加、自减运算符 …… 21
2.7　位运算符 …… 22
2.7.1　按位逻辑运算符 …… 22
2.7.2　移位运算符 …… 24
2.8　逗号运算符和逗号表达式 …… 24
2.9　典型例题分析 …… 25
2.10　实验 2　数据类型、运算符和表达式的使用 …… 28

2.11 习题 …… 30
第 3 章 C 程序中的输入和输出 …… 33
3.1 概述 …… 33
3.2 格式输出函数 printf()和格式输入函数 scanf() …… 33
3.2.1 格式输出函数 printf() …… 33
3.2.2 格式输入函数 scanf() …… 36
3.3 字符输出函数 putchar()和字符输入函数 getchar() …… 37
3.3.1 字符输出函数 putchar() …… 37
3.3.2 字符输入函数 getchar() …… 38
3.4 典型例题分析 …… 39
3.5 实验 3 设计并运行简单的 C 程序 …… 41
3.6 习题 …… 43
第 4 章 C 程序的控制结构 …… 47
4.1 程序算法简介 …… 47
4.1.1 算法的概念 …… 47
4.1.2 算法的表示 …… 48
4.1.3 算法的特性 …… 49
4.2 顺序结构程序设计 …… 49
4.2.1 顺序结构的构成 …… 49
4.2.2 C 程序的基本语句 …… 50
4.2.3 编译预处理命令 …… 51
4.3 关系运算符和关系运算表达式 …… 53
4.3.1 关系运算符 …… 53
4.3.2 关系运算表达式 …… 54
4.4 逻辑运算符和逻辑运算表达式 …… 54
4.4.1 逻辑运算符 …… 54
4.4.2 逻辑运算表达式 …… 55
4.5 选择结构程序设计 …… 56
4.5.1 if 语句 …… 56
4.5.2 if 语句的嵌套 …… 61
4.5.3 switch 语句 …… 65
4.6 循环结构程序设计 …… 67
4.6.1 while 语句 …… 67
4.6.2 do-while 语句 …… 69
4.6.3 for 语句 …… 71
4.6.4 循环的嵌套 …… 74
4.7 continue 语句和 break 语句 …… 76
4.7.1 continue 语句 …… 76
4.7.2 break 语句 …… 77

4.8 典型例题分析 …… 78
4.9 实验 4 选择结构程序设计 …… 85
4.10 实验 5 循环结构程序设计 …… 87
4.11 习题 …… 89
第 5 章 数组 …… 93
5.1 一维数组的定义及应用 …… 93
5.1.1 定义 …… 93
5.1.2 初始化 …… 94
5.1.3 一维数组元素的引用 …… 94
5.2 字符数组与字符串 …… 100
5.2.1 字符数组 …… 100
5.2.2 字符串 …… 101
5.2.3 常用的字符串处理函数 …… 102
5.3 二维数组 …… 106
5.3.1 二维数组的定义和初始化 …… 106
5.3.2 二维数组元素的引用 …… 108
5.4 典型例题分析 …… 110
5.5 实验 6 数组程序设计 …… 115
5.6 习题 …… 117
第 6 章 函数 …… 121
6.1 函数概念 …… 121
6.1.1 概述 …… 121
6.1.2 函数的分类 …… 121
6.2 函数的定义 …… 123
6.3 函数参数和函数的值 …… 123
6.3.1 形式参数和实际参数 …… 123
6.3.2 函数的返回值 …… 125
6.4 函数的调用 …… 126
6.4.1 函数调用的一般形式 …… 126
6.4.2 函数声明 …… 127
6.4.3 函数调用中的值传递和地址传递 …… 128
6.4.4 函数的嵌套调用 …… 130
6.4.5 函数的递归调用 …… 131
6.5 局部变量和全局变量 …… 132
6.5.1 局部变量 …… 132
6.5.2 全局变量 …… 134
6.6 动态存储变量与静态存储变量 …… 136
6.7 内部函数和外部函数 …… 139
6.7.1 内部函数 …… 139

6.7.2 外部函数 …… 140
6.8 典型例题分析 …… 140
6.9 实验 7 函数程序设计 …… 143
6.10 习题 …… 146
第 7 章 指针 …… 150
7.1 指针和指针变量的概念 …… 150
7.1.1 指针的概念 …… 150
7.1.2 指针变量的概念 …… 151
7.2 指针作为函数参数 …… 154
7.3 指针与数组 …… 155
7.3.1 一维数组的指针 …… 155
7.3.2 二维数组的指针 …… 158
7.3.3 字符串的指针 …… 162
7.3.4 指针数组 …… 165
7.4 指针与函数 …… 167
7.4.1 指向函数的指针 …… 167
7.4.2 指针函数 …… 168
7.5 典型例题分析 …… 169
7.6 实验 8 指针程序设计 …… 172
7.7 习题 …… 174
第 8 章 结构体与共用体 …… 178
8.1 结构体 …… 178
8.1.1 结构体类型的定义 …… 178
8.1.2 结构体变量的定义和引用 …… 179
8.1.3 指向结构体类型数据的指针 …… 182
8.1.4 结构体数组 …… 184
8.1.5 结构体与函数 …… 186
8.2 链表 …… 189
8.2.1 动态存储管理 …… 189
8.2.2 链表简介 …… 190
8.2.3 链表的基本操作 …… 191
8.3 共用体 …… 194
8.3.1 共用体变量的定义 …… 194
8.3.2 共用体变量的引用 …… 194
8.4 类型说明符 typedef …… 195
8.5 典型例题分析 …… 196
8.6 实验 9 结构体程序设计 …… 200

8.7 习题 …… 203
第 9 章 文件 …… 210
9.1 概述 …… 210
9.2 文件的读和写 …… 210
9.2.1 文件的打开和关闭 …… 210
9.2.2 读写文件的函数及应用 …… 212
9.2.3 文件读写中的检测函数 …… 219
9.3 典型例题分析 …… 219
9.4 实验 10 文件程序设计 …… 224
9.5 习题 …… 227
附录 …… 233
附录 A 常用字符与 ASCII 代码对照表 …… 233
附录 B C 语言中的关键字 …… 234
附录 C 运算符和结合性 …… 234
附录 D C 库函数 …… 235
附录 E Visual C++6.0 编程环境 …… 238
参考文献 …… 245

第1章 C语言概述

1.1 C语言简介

1.1.1 C语言的产生

20世纪60年代，随着计算机科学的迅速发展，高级程序设计语言Fortran、Algol60等得到了广泛的应用，然而，还缺少一种可以用来开发操作系统和编译程序等系统程序的高级语言，人们只能使用机器语言或汇编语言来编写这些程序，但机器语言和汇编语言存在着不可移植、可读性差、研制软件效率不高等缺点，给编程带来很多不便。于是，在20世纪70年代初，C语言应运而生。

C语言的出现是与UNIX操作系统紧密联系在一起的。它最早源于1968年发表的CPL（Combined Programming Language）语言。C语言的许多重要思想则来源于M. Richards在1969年研制的BCPL（Basic Combined Programming Language）语言，以及在BCPL语言的基础上，由K. Thompson在1970年研制、开发的B语言。K. Thompson用B语言为PDP-7计算机编写了第一个UNIX操作系统。随后D. M. Ritchie于1972年在B语言的基础上开发出C语言，并用C语言完成了在PDP-11计算机上实现的UNIX操作系统。UNIX操作系统的巨大成功也是C语言的巨大成功。

目前，从微型计算机到大型计算机都支持C编译程序。C编译程序不仅能在UNIX操作系统下运行，而且能在DOS、Windows和Linux操作系统下运行。由于C语言本身具有的优越性，它已经成为在各种计算机上、从系统软件设计到工程应用程序开发都能使用的一种高级程序设计语言。

1.1.2 C语言的特点

C语言主要有如下特点：

1）表达能力强且应用灵活。C语言是介于汇编语言和高级语言之间的一种程序设计语言。C语言既有面向硬件和系统、像汇编语言那样可以直接访问硬件的功能，又有高级语言面向用户、容易记忆、便于阅读和书写的优点。

2）程序结构清晰且紧凑。C语言是一种模块化程序设计语言，支持把整个程序分割成若干相对独立的功能模块，并且为模块间的相互调用以及数据传递提供便利，这种模块化的程序结构与系统工程的结构要求相一致。

3）书写简单、易学。

4）目标程序的质量高。C语言提供了一个较大的运算符集合，并且其中大多数运算符可直接翻译成机器代码，因此，由它编写的源程序所生成的机器代码质量较高。

5）可移植性好。C 语言通过调用输入、输出函数实现输入、输出功能。而这些函数属于独立于 C 语言的程序模块库，因此，C 语言本身并不依赖于计算机硬件系统，从而便于在不同的计算机之间实现程序的移植。

6）C 语言是结构化程序设计语言，有利于对程序流程实现有效地控制。

7）C 语言提供了丰富的数据类型。它不仅有字符型、整型以及实型等基本数据类型，而且支持构造类型数据，如数组和结构体等，从而可以适应不同的程序需求。

8）C 语言支持指针和指针变量。允许通过指针和指针变量直接访问内存，从而使程序设计更具灵活性。

由于 C 语言具有上述众多特点，已经成为程序设计的主要语言之一，被广泛应用于微处理机和微型计算机的系统软件和应用软件的开发。

1.2 C 程序的结构及书写格式

几十年来所开发的 C 语言编译程序有多种版本，不同版本 C 语言的功能基本上一致，但也存在少量差异，主要体现在标准函数库中所含函数的种类、调用格式和功能上的稍许差别。本书以 1987 年美国国家标准 C 语言为基础，同时兼顾其他不同版本，以通用性、一致性的内容予以叙述。相关的示例均在 Visual C++6.0 编译环境中调试通过。

1.2.1 C 程序的结构

习惯上，称用 C 语言编写的程序为 C 程序。C 程序通常是由一个或几个函数所组成的。下面是最简单的 C 程序示例。说明：本书中程序代码的左边一列数字序号和冒号是为了方便解释程序行而加上的，不属于程序本身。

【例 1-1】 从键盘输入 3 个整数，输出它们的和。

```
1:  #include <stdio.h>                    /*本程序计算 3 个整数的和*/
2:  void main()
3:  {   int x,y,z,sum;
4:      scanf("%d,%d,%d",&x,&y,&z);
5:      sum＝x+y+z;
6:      printf("sum＝%d\n ",sum);
7:  }
```

运行本例程序时，首先从键盘以输入 3 个整数，然后求这 3 个整数之和，并将结果以十进制整数形式输出显示在屏幕上。

程序说明：

第 1 行：include 文件包含命令，所有的 C 程序均需有此命令。以／*开头，并以*／结束的字符行是程序的注释部分，注释可以出现在程序的任何位置，用以帮助阅读和理解程序。运行程序时，注释部分不被执行。

第 2～7 行：C 语言程序的基本结构为：

```
void main()
{
```

```
        语句行（函数体）
    }
```

其中，main()表示主函数。每个 C 程序必须有一个、而且只能有一个主函数。程序从 main()函数开始执行。main 后面的圆括号()是必需的。

花括号{ }内是主函数的函数体部分，函数体由 C 语言的语句序列构成。C 语言中的语句大致分为两类：一类为说明语句，用来描述数据；另一类为执行语句，用来对数据进行操作。每个语句结束时，必须以分号“;”结尾。

第 3 行：数据类型说明语句。说明 x、y、z、sum 四个变量均为整数类型的数据。

第 4 行：执行语句。scanf()是 C 语言提供的格式化输入函数。表示从键盘输入三个整数分别赋给变量 x、y、z。其中的%d 表示整数格式。

第 5 行：执行语句。将 x+y+z 的结果赋值给变量 sum。

第 6 行：执行语句。printf()是 C 语言提供的格式化输出函数。其中，\n 表示输出结束后将光标移至下一行的开始处，即换行。

运行结果：

输入：1，2，3 <回车>

输出：sum = 6

【例 1-2】 采用模块结构，改写例 1-1 的程序。

```
1:   #include <stdio.h>
2:   int add(int x,int y,int z )
3:       {      return(x+y+z);
4:       }
5:   void main()
6:     {
7:         int x,y,z;
8:         printf("Please Input Three Integers:\n ");
9:         scanf("%d,%d,%d ",&x,&y,&z);
10:        printf("sum＝%d\n ",add(x,y,z));
11:    }
```

程序说明：

第 2～4 行：定义函数 add()，括号中的 x、y、z 称为形式参数，该函数的功能是返回 3 个整数之和。

第 3 行：将 x+y+z 的结果返回给调用函数 add()的函数 main()。

第 5～11 行：主函数。

第 8 行：人机对话，显示必要的信息，提示用户输入 3 个整数。

第 10 行：输出结果。其中通过 add(x,y,z)调用函数 add()，调用时，将用户输入的 x、y、z 的值传递给相应的形式参数，调用结束时，返回结果，由输出函数 printf()显示结果。

运行结果：

显示：Please Input Three Integers:

输入：1，2，3<回车>

输出：sum = 6

与例 1-1 的程序比较，例 1-2 将 3 个整数求和的计算从主函数中分离出去，由一个函数 add()实现这一功能，而主函数 main()只承担输入数据、调用函数 add()和输出结果的功能，称这种程序结构为模块化程序结构，相应地，称这种程序设计方法为模块化程序设计方法。模块化程序设计的特点是由不同的函数实现不同的功能，在主函数的统一调度下，实现既定目标。

C 程序的模块化程序结构如图 1-1 所示。

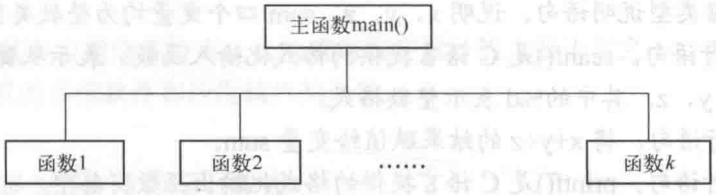

图 1-1　C 程序的模块化程序结构

习惯上称图中的函数 1、……、函数 k 为子函数。C 语言规定，主函数可以调用子函数，各子函数之间也可以互相调用，但子函数不可以调用主函数。

采用模块化程序设计的优点是：

- 程序结构清晰、层次分明。
- 易于调试。
- 便于删除或添加功能。
- 符合系统设计的要求。

1.2.2　C 程序的书写格式

C 程序的书写须遵循下列规则：

1）用 C 语言书写程序时较为自由，既可以一行写一个或多个语句，也可以一个语句分几行来写。

2）C 语言规定了若干有特定意义、为 C 语言专用的单词，称为关键字，如例 1-1 中的 main、int、scanf、printf 和例 1-2 中的 return 都是 C 语言的关键字。C 语言规定关键字必须使用小写字母。习惯上，书写 C 程序时均使用小写英文字母。

3）为了看清 C 程序的层次结构，便于阅读和理解程序，C 程序一般都采用缩进格式的书写方法。缩进格式要求在书写程序时，不同结构层次的语句，从不同的起始位置开始，同一结构层次中的语句，缩进同样个数的字符位置。从第 4 章开始，读者能从相关的例子中体会到采用缩进格式书写 C 程序的益处。

4）为了便于阅读和理解程序，应当在程序中适当地添加一些注释行。

1.3　C 程序的开发过程

程序员所编写的源程序（以.c 为文件扩展名）是无法直接运行的，必须由 C 编译程序对其进行编译，最终生成机器代码才能为计算机所识别并执行。C 编译程序首先对源程序进行

语法检查，若没有发现错误，编译后将产生目标代码，并生成目标文件（以.obj 为文件扩展名）。若编译程序发现源程序有错误，则输出错误信息，此时，程序员应该对程序进行修改，纠正错误后，再进行编译，直到编译正确为止。需要指出的是：C 编译程序无法检查出用户程序中的算法错误，这类错误只能由用户根据实际问题以及凭经验加以判断和纠正。

经编译程序编译后产生的目标文件还是不能直接在计算机上运行，它仅仅是一个内存地址浮动的程序模块，还需要将程序重新定位在内存中确定的绝对地址上，此外，还必须将目标代码同源程序中所调用的标准函数库文件（如：scanf()、printf()）的目标代码结合起来。这个过程称为“连接”，由 C 语言的连接程序完成，最后生成可执行文件（以.exe 为文件扩展名）。该可执行文件才可以在 DOS 系统下直接运行。

因此，C 程序整个开发过程包括 4 部分：编辑源程序、编译程序、连接程序和运行程序。其示意图如图 1-2 所示。

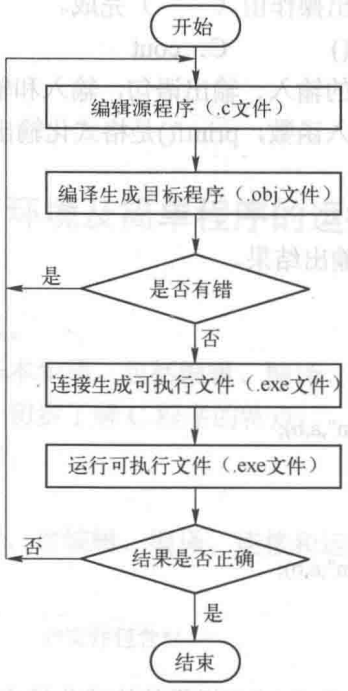

图 1-2　C 程序的开发过程示意图

1.4　典型例题分析

【例 1-3】 以下叙述正确的是（　　）。

A．组成 C 程序的是函数。

B．组成 C 程序的是 main()函数。

C．C 程序总是从第一个函数开始执行。

D．C 程序中，注释只能位于一条语句之后。

解析：对 C 程序应明确：C 程序的基本单位是函数，C 程序由一个或几个函数构成，其

中必须包含 main()主函数。C 程序书写格式自由，每个函数在整个程序中的位置任意，main()主函数不一定出现在程序的开始处，但不管 main()主函数位于程序的何处，C 程序总是从 main()函数开始执行，函数体必须以“{”开始，以“}”结束。程序的注释部分应包括在/*...*/ 之间，/和*之间不允许留有空格，/*和*/应当成对出现；注释部分允许出现在程序的任何位置，它对程序的执行不产生任何影响。

由此可知，答案是 A。

【例 1-4】 C 源文件的扩展名为（　　）。

A．cpp　　B．txt　　C．c　　D．exe

解析：C 源程序的扩展名为 c，C++源程序的扩展名为 cpp，文本文件的扩展名为 txt，源程序经过编译、连接后得到可执行文件的扩展名为 exe。

由此可知，答案是 C。

【例 1-5】 在 C 语言中，输出操作由（　　）完成。

A．scanf()　　B．printf()　　C．cout　　D．输出语句

解析：C 语言没有提供专门的输入、输出语句，输入和输出都是由 C 语言提供的库函数来完成，其中 scanf()是格式化输入函数，printf()是格式化输出函数，而 cout 是 C++中的标准输出流对象。

由此可知，答案是 B。

【例 1-6】 写出下列程序的输出结果。

```
1：#include <stdio.h>
2：void main()
3：{    int a=2,b=5;
4：     printf("a=%d,b=%d\n",a,b);
5：     a=a+b;
6：     b=a-b;
7：     a=a-b;
8：     printf("a=%d,b=%d\n",a,b);
9：}
```

解析：

第 4 行：执行语句。printf()是 C 语言提供的格式化输出函数。其中，双引号中的%d 表示以整数格式输出，\n 表示输出结束后换行，其他字符应原样输出。

第 5 行：赋值语句。执行之后变量 a 的新值为 a 和 b 之和 7。

第 6 行：赋值语句。变量 b 值是变量 a 的原来值 2。

第 7 行：赋值语句。变量 a 的最新值是变量 b 的原来值 5。

由此可知，通过 5、6、7 三条赋值语句，使变量 a、b 的值交换。所以运行结果为：

```
a=2,b=5
a=5,b=2
```

【例 1-7】 写出下列程序的输出结果。

```
1：#include <stdio.h>
```

```
2：void main()
3：{   int a=2,b=5,t;
4：    printf("a=%d,b=%d\n",a,b);
5：    t=a;
6：    a=b;
7：    b=t;
8：    printf("a=%d,b=%d\n",a,b);
9：}
```

解析：第 5 行：赋值语句。执行之后变量 t 的值为 a 的值 2。

第 6 行：赋值语句。变量 a 的新值是变量 b 的值 5。

第 7 行：赋值语句。变量 b 的新值是变量 t 的值，即变量 a 的原来值 2。

由此可知，通过 5、6、7 三条赋值语句，也可以使变量 a、b 的值交换。所以运行结果为：

```
a=2,b=5
a=5,b=2
```

1.5 实验 1　C 程序运行环境及简单程序的运行

一、实验目的与要求

1）熟悉 C 语言集成编译环境。

2）掌握运行一个 C 程序的基本步骤，包括编辑、编译、连接和运行。

3）通过运行简单的 C 程序，初步了解 C 程序的特点。

4）理解一些最基本的 C 语句。

二、实验内容

1．下面是一个简单的 C 程序，请编辑、编译、连接和运行该程序，观察并记下屏幕的输出结果。

```
#include<stdio.h>                    /*文件包含*/
void main()
{
    int a,b;                         /*定义整型变量 a,b*/
    a=10;                            /*赋值语句，将 10 赋给变量 a*/
    b=a+20;                          /*赋值语句，将 a+10 赋给变量 b*/
    printf ("b=%d\n",b);             /*将变量 b 的值以%d 十进制整数形式输出*/
}
```

【使用 Visual C++实验步骤】

第 1 步：进入 Visual C++环境后，打开“文件”菜单，执行“新建”命令。

第 2 步：在“新建”对话框中选择“文件”选项卡，然后选择 C++ Source File 选项。

第 3 步：在右边的目录文本框中输入准备编辑的源程序文件的存储路径，在文件文本框中输入准备编辑的 C 源程序文件名（如：sy1_1.c）。注意扩展名是.c，然后按“确定”按钮。

第 4 步：在光标闪烁的程序编辑窗口输入上面 C 程序（注意：/*　*/之间的内容为程序注释部分，不执行），程序输入完毕单击“文件”→“保存”，或单击工具栏上的“保存”按钮，也可以用〈Ctrl+S〉快捷键来保存文件。

第 5 步：选择菜单“编译”→“编译”命令，或单击工具栏上的“编译”图标，也可以按〈Ctrl+F7〉快捷键，开始编译。观察调试信息窗口输出编译的信息，如果有错，则修改后再编译，直至编译信息为：0 error(s)，0 warning(s)，表示编译成功。

第 6 步：单击〈F7〉键或工具栏图标，生成应用程序的.EXE 文件（如 sy1_1.exe）。

第 7 步：运行程序观察结果。选择菜单“编译”→“执行”，或单击工具栏上的执行图标，也可以使用〈Ctrl+F5〉快捷键。

2．改错题

1）下列程序的功能为：计算 x+y 的值并将结果输出。请纠正程序中存在错误，使程序实现其功能。

```
#include<stdio.h>
void main()
{
    int x,y;
    x=10，  y=20
    sum=x+y;
    printf("x+y=%d",sum)
    printf("\n");
}
```

2）下面程序的功能是：求半径为 r 的圆面积。请纠正程序中存在错误，使程序实现其功能。

```
#include<stdio.h>
void main()
{
    float r,area;                          /*定义浮点型变量 r,area*/
    printf("Please input r (r>0):");
    scanf("%f",r);                         /*%f 是浮点数格式控制符*/
    area=3.1416r*r;
    printf("r=%6.2f\n",r);
    printf("area=%d\n",area);
}
```

3．程序填空题

1）下面程序的功能是：从键盘输入两个整数，输出这两个整数的和。请根据注释信息填写完整程序，使程序实现其功能。运行程序，观察并记下屏幕的输出结果。

```
#include<stdio.h>
void main()
{
    _____                                  /* 定义整型变量 a，b，sum */
```