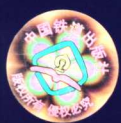


21世纪大学计算机教育系列规划教材

# C 语言程序设计

刘明才 编著



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

第1章 绪论 1

1.1 C语言的发展及其特点 1

1.2 C语言的应用 2

1.3 本书的组织 3

1.4 习题1 4

1.5 本章小结 4

1.6 本章参考文献 4

1.7 本章思考题 4

1.8 本章实验 4

1.9 本章小结 4

1.10 本章习题 4

1.11 本章实验 4

1.12 本章小结 4

1.13 本章参考文献 4

1.14 本章思考题 4

1.15 本章实验 4

1.16 本章小结 4

1.17 本章参考文献 4

1.18 本章思考题 4

1.19 本章实验 4

1.20 本章小结 4

1.21 本章参考文献 4

1.22 本章思考题 4

1.23 本章实验 4

1.24 本章小结 4

1.25 本章参考文献 4

1.26 本章思考题 4

1.27 本章实验 4

1.28 本章小结 4

1.29 本章参考文献 4

1.30 本章思考题 4

1.31 本章实验 4

1.32 本章小结 4

1.33 本章参考文献 4

1.34 本章思考题 4

1.35 本章实验 4

1.36 本章小结 4

1.37 本章参考文献 4

# C语言程序设计

第1版



清华大学出版社

21 世纪大学计算机教育系列规划教材

# C 语言程序设计

刘明才 编著

**中国铁道出版社**  
CHINA RAILWAY PUBLISHING HOUSE

---

## 内 容 简 介

本书主要介绍了C语言程序设计的基础知识、基本数据类型与常用库函数、各种运算符和表达式、控制结构和语句、数组、函数、编译预处理、指针、结构体与共用体、文件及C++程序设计基础知识等。

本书是学习C语言程序设计的基础教材,通过循序渐进的内容安排,通俗易懂的讲解,使读者能够掌握C语言的基本内容,并具有一定的程序设计能力。

本书按基础课程要求编写,可作为高等院校相关专业的C语言程序设计课程的教材,也可作为计算机等级考试的教学用书。

### 图书在版编目(CIP)数据

C语言程序设计/刘明才编著. —北京:中国铁道出版社, 2007. 1

(21世纪大学计算机教育系列规划教材)

ISBN 978-7-113-07627-6

I. C... II. 刘... III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2007)第008045号

书 名: C语言程序设计

作 者: 刘明才

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 秦绪好

责任编辑: 苏 茜 康珊珊 王慧亮

封面设计: 付 巍

封面制作: 白 雪

责任校对: 王春霞

印 刷: 北京市兴顺印刷厂

开 本: 787×1092 1/16 印张: 16 字数: 384千

版 本: 2007年1月第1版 2007年1月第1次印刷

印 数: 1~5 000册

书 号: ISBN 978-7-113-07627-6/TP·2225

定 价: 22.00元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

C 语言是一种非常出色的程序设计语言，它结构严谨、数据类型完整、语句简练灵活、运算符丰富。C 语言具有很强的实用性，既可以用于编写系统软件，又可以用于编写各种应用软件。虽然面向对象程序设计语言在今天广泛使用，但 C 语言仍然是学习程序设计的基础语言。现在，国内几乎所有的高等院校都开设了“C 语言程序设计”课程。当今人们对计算机知识的需求使得 C 语言不仅成为计算机专业学生的必修课程，也成为广大非计算机专业学生和计算机爱好者首选的程序设计语言，而且全国和各省的计算机等级考试也都将 C 语言列为重要的考试内容之一。

本书面向初学程序设计者，只要是略有计算机基础知识的人都能较容易地学会用 C 语言编写程序。在编写教材过程中，作者力求使教材体现如下特点。

- 对 C 语言中的重点、难点进行分解，将重点、难点分散编排，使学生在过程中循序渐进、平滑过渡。
- 对 C 语言中生僻、不常用的内容不作过多的描述；对于在实践中使用较多、需牢固掌握的内容进行了详细的叙述，并给出了大量的例子。
- 本书在介绍 C 语言的基本知识和语法的同时，还强调读者编程风格的形成，有意识地训练读者逐步养成一个良好的程序书写习惯和程序设计风格。
- 本书所有例子均在 Turbo C 环境下调试通过，也能在 Visual C++ 环境下运行。

为方便学习和加强实验教学，同时编写了该书的配套用书《C 语言程序设计习题解答与实验指导》。

本书共分 11 章。第 1 章介绍了 C 程序的基础知识与 Turbo C 的简单使用方法。第 2 章介绍了基本数据类型、常用的输入/输出函数。第 3 章介绍了各种运算符和表达式。第 4 章介绍了 C 语言的控制结构（顺序结构、选择结构和循环结构）。第 5 章介绍了数组的使用及常用字符串处理函数。第 6 章介绍了函数的使用及变量的存储类别。第 7 章介绍了编译预处理命令。第 8 章介绍了指针的使用。第 9 章介绍了结构体和共用体的使用、链表及其操作。第 10 章介绍了文件的类型和操作。第 11 章对 C++ 程序设计做了简单介绍。

本书作者长期从事高等学校“C 语言程序设计”课程的教学工作，在总结多年的“C 语言程序设计”课程的教学经验和教改实践的基础上，编写了这本教材。由于作者水平有限，书中不足之处在所难免，恳请广大读者批评指正。

编者

2006 年 11 月

<b>第 1 章 C 语言概述</b> .....	1
1.1 计算机语言与程序.....	1
1.2 C 语言的发展历史.....	2
1.3 C 语言的特点.....	3
1.4 C 语言程序设计初步.....	4
1.4.1 简单的 C 程序实例.....	4
1.4.2 C 程序结构.....	6
1.5 C 程序在计算机上执行.....	7
1.5.1 C 程序执行的步骤.....	7
1.5.2 用 Turbo C 2.0 运行 C 程序的步骤.....	8
习题 1.....	10
<b>第 2 章 基本数据类型与常用库函数</b> .....	12
2.1 字符集与标识符.....	12
2.2 数据类型与基本数据类型.....	13
2.2.1 数据类型.....	13
2.2.2 基本数据类型.....	14
2.3 常量与变量.....	15
2.3.1 常量.....	15
2.3.2 符号常量.....	18
2.3.3 变量.....	18
2.4 数据的输入/输出函数.....	19
2.4.1 字符输入/输出函数.....	19
2.4.2 格式输出函数 printf.....	20
2.4.3 格式输入函数 scanf.....	24
2.5 常用库函数.....	27
2.5.1 常用数学函数.....	28
2.5.2 常用字符函数.....	29
2.5.3 其他常用函数.....	30
习题 2.....	30
<b>第 3 章 运算符和表达式</b> .....	33
3.1 算术运算符和算术表达式.....	33
3.2 自增、自减运算.....	36
3.3 位运算.....	38
3.4 赋值运算.....	41

3.5	关系表达式与逻辑表达式	42
3.5.1	关系运算符	43
3.5.2	逻辑运算符	44
3.6	条件表达式与逗号表达式	46
3.6.1	条件表达式	46
3.6.2	逗号表达式	47
	习题 3	47
<b>第 4 章</b>	<b>C 语言的控制结构</b>	<b>51</b>
4.1	算法与程序	51
4.1.1	算法	51
4.1.2	程序	52
4.2	顺序结构	53
4.3	选择结构	54
4.3.1	if 语句	54
4.3.2	switch 语句	59
4.4	循环结构	62
4.4.1	while 语句	62
4.4.2	do...while 语句	63
4.4.3	for 语句	65
4.4.4	单重循环程序举例	67
4.4.5	多重循环结构	69
4.4.6	break 语句	71
4.4.7	continue 语句	72
4.4.8	goto 语句	72
	习题 4	73
<b>第 5 章</b>	<b>数组</b>	<b>78</b>
5.1	一维数组	78
5.1.1	一维数组的定义	78
5.1.2	一维数组的存储形式	79
5.1.3	一维数组的初始化	79
5.1.4	一维数组的引用	79
5.1.5	一维数组的应用举例	80
5.2	二维数组	83
5.2.1	二维数组的定义	83
5.2.2	二维数组的存储形式	83
5.2.3	二维数组的初始化	84
5.2.4	二维数组的引用	85
5.2.5	二维数组的应用举例	85

5.3 字符数组 .....	87
5.3.1 字符数组的定义 .....	87
5.3.2 字符数组的初始化 .....	87
5.3.3 字符数组的输入/输出 .....	88
5.3.4 常用字符串处理函数 .....	89
5.3.5 二维字符数组 .....	91
5.3.6 字符串应用举例 .....	92
习题 5 .....	94
<b>第 6 章 函数</b> .....	<b>99</b>
6.1 函数概述 .....	99
6.2 函数定义和函数调用 .....	100
6.2.1 函数定义 .....	100
6.2.2 return 语句 .....	102
6.2.3 函数调用 .....	102
6.2.4 函数声明 .....	103
6.2.5 函数间的参数传递 .....	104
6.3 函数的嵌套调用和递归调用 .....	106
6.3.1 函数的嵌套调用 .....	106
6.3.2 函数的递归调用 .....	107
6.4 局部变量和全局变量 .....	111
6.4.1 局部变量 .....	111
6.4.2 全局变量 .....	112
6.4.3 如何用 Turbo C 运行多文件的程序 .....	115
6.5 变量的存储类别 .....	115
6.5.1 auto (自动) 变量 .....	116
6.5.2 static (静态) 变量 .....	116
6.5.3 register (寄存器) 变量 .....	117
6.6 内部函数和外部函数 .....	118
习题 6 .....	118
<b>第 7 章 编译预处理</b> .....	<b>123</b>
7.1 宏定义 .....	123
7.1.1 不带参数的宏定义 .....	123
7.1.2 带参数的宏定义 .....	124
7.1.3 终止宏定义 .....	126
7.2 文件包含 .....	126
7.3 条件编译 .....	128
习题 7 .....	129



<b>第 8 章 指针</b> .....	132
8.1 指针的基本概念.....	132
8.1.1 变量的地址及指针.....	132
8.1.2 指针变量.....	132
8.1.3 指针运算.....	135
8.1.4 指针作为函数的参数.....	136
8.2 指针与一维数组.....	139
8.2.1 一维数组元素的表示.....	139
8.2.2 数组名作为函数的参数.....	142
8.2.3 数组元素指针作为函数的参数.....	143
8.3 指针与二维数组.....	144
8.3.1 二维数组中的指针.....	144
8.3.2 指向二维数组元素的指针变量.....	146
8.3.3 二维数组的行指针作函数参数.....	148
8.4 指针与字符串.....	150
8.4.1 字符串的表示形式.....	150
8.4.2 字符串指针作函数参数.....	151
8.5 指针与函数.....	152
8.5.1 指向函数的指针.....	152
8.5.2 返回指针的函数.....	153
8.6 指针数组与多级指针.....	154
8.6.1 指针数组的概念.....	154
8.6.2 多级指针的概念.....	155
8.6.3 指针数组作 main 函数的形参.....	156
习题 8.....	157
<b>第 9 章 结构体与共用体</b> .....	162
9.1 结构体类型的定义.....	162
9.2 结构体类型变量.....	163
9.2.1 结构体类型变量的定义.....	163
9.2.2 结构体变量的引用.....	164
9.3 结构体数组.....	166
9.3.1 结构体数组的定义与初始化.....	166
9.3.2 结构体数组的引用.....	166
9.4 结构体和指针.....	167
9.5 结构体和函数.....	169
9.5.1 结构体作函数参数.....	169
9.5.2 返回结构体的函数.....	171
9.6 链表.....	173

9.6.1	简单链表.....	173
9.6.2	处理动态链表所需的库函数.....	174
9.6.3	单向链表的基本操作.....	175
9.7	共用体.....	182
9.8	枚举.....	184
9.9	用 typedef 定义类型.....	186
习题 9	.....	187
<b>第 10 章</b>	<b>文件.....</b>	<b>191</b>
10.1	文件的基本概念.....	191
10.2	文件的打开与关闭.....	191
10.2.1	文件类型指针.....	191
10.2.2	文件的打开.....	192
10.2.3	文件的关闭.....	193
10.3	文件的读写.....	193
10.3.1	字符读写函数 fgetc 和 fputc.....	194
10.3.2	字符串读写函数 fgets 和 fputs.....	196
10.3.3	数据块读写函数 fread 和 fwrite.....	197
10.3.4	格式化读写函数 fscanf 和 fprintf.....	198
10.4	文件的随机读写.....	199
10.4.1	文件定位.....	199
10.4.2	文件的随机读写.....	200
习题 10	.....	201
<b>第 11 章</b>	<b>C++程序设计基础.....</b>	<b>204</b>
11.1	C++语言简介.....	204
11.2	C++程序简介.....	205
11.2.1	一个简单的 C++程序.....	205
11.2.2	用 Visual C++运行一个简单的 C++程序.....	205
11.3	C++对 C 的扩充.....	209
11.3.1	C++的输入/输出.....	209
11.3.2	函数的重载.....	211
11.3.3	带缺省参数的函数.....	213
11.3.4	内联函数.....	214
11.3.5	变量的引用类型.....	214
11.3.6	作用域运算符.....	217
11.3.7	const 关键字.....	218
11.3.8	动态分配/撤销内存的运算符 new 和 delete.....	219
11.4	类和对象.....	221
11.4.1	类和对象的定义.....	221

11.4.2	构造函数.....	225
11.4.3	析构造函数.....	227
11.5	继承与派生.....	228
11.5.1	继承与派生的概念.....	228
11.5.2	建立派生类的方法.....	229
11.5.3	派生类的三种继承方式.....	230
11.5.4	派生类的构造函数和析构造函数.....	232
11.5.5	多继承.....	234
习题 11	.....	236
附录 A	C 语言中的关键字.....	239
附录 B	字符与 ASCII 码对照表.....	240
附录 C	运算符的优先级与结合性.....	241
附录 D	常用库函数.....	242
参考文献	.....	246

# 第 1 章 C 语言概述

C 语言是国际上广泛流行的计算机高级语言，既可以用来编写系统软件，又可以用来编写应用软件。首先介绍计算机语言和程序执行过程。

## 1.1 计算机语言与程序

### 1. 程序

通常，完成一项复杂的任务，需要进行一系列的具体工作。这些按一定顺序安排的工作即操作序列，称为程序（program）。例如，学校里开会的程序步骤如下。

- (1) 宣布大会开始。
- (2) 介绍出席大会的领导。
- (3) 领导讲话。
- (4) 宣布大会结束。

可见，程序的概念是很普遍的。对于计算机来说，计算机要完成某种数据处理任务，我们可以设计计算机程序，即规定一组操作步骤，使计算机按该操作步骤执行，完成该数据处理任务。在为计算机设计程序时，必须用特定的计算机语言描述。用计算机语言设计的程序，即为计算机程序。程序就是计算机为完成某一个任务所必须执行的一系列指令的集合。

### 2. 计算机语言

计算机系统所能接受、理解和执行的语言称为计算机语言（computer language）。计算机语言主要分为 3 种类型：机器语言、汇编语言和高级语言。

#### (1) 机器语言

机器语言（machine language）中的基本元素是“0”和“1”。由“0”和“1”组成的一个二进制编码，就表示一条机器指令（machine instruction），它使计算机完成一个简单的操作。不同型号的计算机系统，其指令的表示形式和范围是不同的。每种型号的机器指令，都有自己的编码格式，组成自己的指令集合，称为计算机的指令系统，一般由数百条指令组成，每条指令使计算机完成一种简单的操作。用机器指令编写的程序，就是机器语言程序，它是计算机唯一能够直接识别并执行的程序。

#### (2) 汇编语言

汇编语言（assembly language）是在机器语言基础上，引入了助记符、符号数来表示机器指令中的操作码和操作数地址。助记符通常用英文单词缩写，便于记忆。使用汇编语言编写的程序称为汇编语言源程序（source program）。

#### (3) 高级语言

高级语言（high-level language）是一种用户容易理解和掌握的计算机语言，它与人们习惯的自然语言和数学语言很接近，相对汇编语言来说容易掌握和使用。使用高级语言编写的

程序称为高级语言源程序。随着计算机技术的发展,用于各种不同领域的高级语言相继问世。C 语言,以它自身特有的优点成为编写系统软件和应用软件较理想的计算机高级语言。

### 3. 程序执行

并不是所有用计算机语言编写的程序,计算机都能立即执行。计算机只能直接识别执行机器语言程序或称目标程序(object program)。而用汇编语言或高级语言编写的程序,都不能直接在计算机上执行。汇编语言源程序需要通过一个称为汇编程序(assembly)的软件,将汇编语言源程序翻译成计算机可以直接识别运行的机器语言程序。

对于用高级语言编写的源程序,需要经过一个称为“翻译程序”的软件将其翻译成计算机能够直接识别的机器语言程序,才能执行。通常源程序中的每一条语句要翻译成一组机器指令。翻译程序有两种,即编译程序(compiler)和解释程序(interpreter)。每种高级语言都有自己的翻译程序,用 C 语言编写的源程序,首先通过 C 编译程序,将整个源程序翻译成机器语言程序,然后还要通过连接程序生成可执行程序,才能在机器上运行。C 语言源程序的执行过程,如图 1-1 所示。

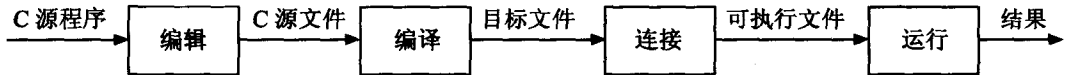


图 1-1 C 语言源程序的执行过程

## 1.2 C 语言的发展历史

C 语言是为了描述 UNIX 操作系统,由美国贝尔实验室在 B 语言的基础上发展起来的。1969 年,贝尔实验室的研究人员 Ken Thompson 和 Dennis M. Ritchie 合作用汇编语言编制了 UNIX 操作系统。1970 年, Ken Thompson 为了提高 UNIX 的可读性和可移植性,在一种叫做 BCPL 语言的基础上开发了一种新的语言,称为 B 语言。由于 B 语言存在一些缺点,无法支持多种数据类型,因此没有流行起来。

1972 年, Denis M. Ritchie 在 B 语言的基础上设计出了 C 语言。1973 年, Ken Thompson 和 Dennis M. Ritchie 合作将原来用汇编语言编写的 UNIX 操作系统 90% 以上的部分用 C 语言改写。与此同时, C 语言的编译程序也被移植到 IBM 360/370、Honeywell 11 以及 VAX-11/780 等多种计算机上,迅速成为应用最广泛的系统程序设计语言。

1978 年, Brian W. Kernighan 和 Dennis M. Ritchie (K&R) 合著出版了一本名为《The C Programming Language》(《C 程序设计语言》)的书,这本书流行很广,被公认为 C 语言的标准版本,称为标准 C。

1983 年,美国国家标准化协会 ANSI (american national standard institute) 根据 C 语言的发展制定了新的标准,称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展。国际标准化组织 ISO (international standard organization) 于 1990 年采用了一个 C 标准 (ISO C)。ISO C 和 ANSI C 实质上是同一个标准。ANSI/ISO 标准的最终版本通常被称为 C90。然而,因为 ANSI 版本是首先出现的,所以人们通常使用 ANSI C 这一术语。

20 世纪 90 年代,虽然大多数程序员都在忙于 C++ 标准的开发,但 C 语言并没有停滞不

前。新的标准在不断开发，最终形成了1999年的C语言标准，通常称为C99。C99基本保留了C90的全部特性，C99标准化委员会致力于两个主要方面：增加一些数据库函数和开发一些专用的但高度创新的新特性。例如，可变长度的数组和restrict指针修饰符。这些改进再一次将C语言推到了计算机语言开发的前沿。结果是C99的修改保持了C语言的本质特性，C语言继续是一种简短、清楚和高效的语言。

## 1.3 C语言的特点

C语言是一种通用、灵活、结构化和使用普遍的计算机高级语言，能完成用户想实现的任何任务，既可以用来编写系统软件，又可以用来编写应用软件。和其他高级语言比较，C语言有如下主要特点。

### 1. C语言简洁紧凑

C语言一共有32个关键字，9种控制语句，书写形式自由，使用方便、灵活。

### 2. C语言介于汇编语言与高级语言之间

C语言既像汇编语言那样允许直接访问物理地址，能进行位运算，能实现汇编语言的大部分功能，直接对硬件访问；又具有高级语言的面向用户、容易记忆、容易学习和易于书写的特点。

### 3. C语言是一种结构化语言

C语言具有结构化语言所规定的3种基本结构。C语言用函数作为结构化程序设计的实现工具，实现程序的模块化。

### 4. C语言有丰富的数据类型

C语言具有现代化高级语言的各种数据类型；用户能自己扩充数据类型，实现各种复杂的数据结构，完成用于具体问题的数据描述。尤其是指针类型，它是C语言的一大特色，灵活的指针操作，能够高效处理各种数据。

### 5. C语言有丰富的运算符

ANSI C提供了34种运算符，灵活使用这些运算符，可以编写各种各样的表达式，表达能力很强，表达式的学习是学习C语言的重点和难点之一。

### 6. C语言具有较高的移植性

C语言在可移植性方面处于领先地位。在C语言中，没有专门与硬件有关的输入/输出语句，程序的输入/输出通过调用库函数实现，使C语言本身不依赖于硬件系统，编写出的程序具有良好的可移植性。

### 7. 灵活性

C语言的语法限制不太严格，对程序员没有施加过多的限制，程序设计能够很自由地进行，这给程序设计带来了很大的灵活性。

### 8. 缺点

C语言的某些优秀特性在给用户带来方便的同时，也可能给用户带来风险。例如，如果对C语言中的指针使用不当，可能带来严重后果（如非法访问内存地址等）。另外，C语言的简洁性，使用户可能编写出很多难以理解的程序，降低了可读性。

虽然前面介绍了 C 语言的主要特点,但对于初学者而言,可能暂时无法明白,只有在使  
用 C 语言的过程中,或者在学习了其他高级语言并与之比较后,才能真正理解这些特点。

## 1.4 C 语言程序设计初步

### 1.4.1 简单的 C 程序实例

下面介绍几个简单的 C 语言程序,并对程序作适当的解释。读者通过这些简单的例子,  
可以对 C 语言的语法和程序结构有一个感性的认识。

用 C 语言编写的程序,称为 C 语言源程序 (source program),简称为 C 程序。C 程序一  
般由一个或多个文件组成,为了表示这些文件中存放的是 C 程序,它们必须以.c 作为文件的  
后缀 (扩展名)。

**【例 1.1】**在屏幕上输出如下内容:

```
Hello,Everyone  
Study hard!
```

程序如下:

```
#include<stdio.h>  
void main()  
{ printf("Hello,Everyone\nStudy hard!");  
}
```

运行情况如下:

```
Hello,Everyone  
Study hard!
```

程序说明:

(1) 这是一个简单的 C 程序。编译预处理命令“# include<stdio.h>”的作用是将“stdio.h”  
文件的内容嵌入到程序中,使输入/输出能正常执行。

(2) main 是函数名,后面必须有一对圆括号。一个完整的 C 程序必须有一个 main 函数  
(即主函数)。一个 C 程序总是从 main 函数开始执行的。main 前面的 void 指明了 main 函数  
的返回值类型,void 表示空,就是不返回任何值。

(3) void main()下面有一对花括号,花括号内的部分称为函数体,由若干语句组成,用  
于完成函数的功能。

(4) printf 是 C 库函数中的一个函数,其作用是在屏幕上输出指定的内容。此程序中输  
出“Hello,Everyone\nStudy hard!”字符串,其中“\n”是输出一个换行,使后面的“Study hard!”  
字符串在下一行输出。C 语言规定每个语句以分号 (;) 结束,分号是语句不可缺少的部分。  
第一行的“#include<stdio.h>”并不是 C 语句,在末尾不加分号,它是一条预处理命令,在编  
译之前执行。

**【例 1.2】**一个简单的加法计算程序,求两个整数的和。

程序如下:

```
#include<stdio.h>  
void main()
```

```

{ int x,y,sum; /* 定义存放整数的变量 x、y、sum */
  printf("Input a integer:"); /* 显示提示信息"Input a integer:" */
  scanf("%d",&x); /* 给变量 x 输入一个整数 */
  printf("Input another integer:"); /* 显示提示信息"Input another integer:" */
  scanf("%d",&y); /* 给变量 y 输入一个整数 */
  sum=x+y; /* 计算 x+y, 将把结果赋给 sum */
  printf("sum=%d\n",sum); /* 输出 sum 的值 */
}

```

运行情况如下:

```

Input a integer: 3✓
Input another integer: 5✓
sum=8

```

程序说明:

(1) 整个程序由一个 main 函数组成。

(2) “int x, y, sum;”是变量定义语句,它告诉计算机,程序中用到整型变量 x、y、sum,并为它们分配存储单元。

(3) “scanf(“%d”,&x);”是输入语句。其中,scanf 是 C 的库函数中的一个函数,其作用是给指定的变量输入数据,“%d”是格式说明符,“&x”表示 x 的地址,该语句表示给 x 输入一个整数。当程序执行到该语句时,系统处于等待状态,等待用户从键盘输入一个整数,然后按回车键,该整数就赋给了相应的变量。在本书中,用符号“✓”表示回车键(【Enter】键),如“3✓”表示从键盘输入 3,再按回车键。

(4) “sum=x+y;”是计算两个变量 x 与 y 的和,并将结果存放到变量 sum 中。

(5) “printf(“sum=%d\n”,sum);”中的“%d”是格式说明符,表示在该位置输出 sum 的值,如本例中输出 sum=8。

(6) /\* ..... \*/ 是注释,在程序执行中不起任何作用,只是增加程序的可读性。任何文字都可用“/\*”和“\*/”括起来,作为程序的注释部分。

**【例 1.3】**输入 3 个数,求其中最大的数。

程序如下:

```

#include<stdio.h>
float max(float x,float y) /* 定义求两个实数最大值的函数 max */
{ float z; /* 定义存放实数的变量 z */
  if(x<y) z=y;else z=x; /* 比较 x 和 y,将较大者赋给 z */
  return z; /* 返回 z 的值 */
}
void main()
{ float a,b,c,d,e; /* 定义存放实数的变量 a、b、c、d、e */
  printf("a,b,c=?\n"); /* 输出提示信息"a,b,c=?" */
  scanf("%f%f%f",&a,&b,&c); /* 输入 3 个实数,分别赋给 a、b、c */
  d=max(a,b); /* 调用函数 max,求 a、b 较大者 */
  e=max(d,c); /* 调用函数 max,求 d、c 较大者 */
  printf("max=%f\n",e);
}

```



运行情况如下:

```
a, b, c=?
3 8 -6✓
max=8.000000
```

程序说明:

- (1) max 函数的作用是取两个实数  $x$  和  $y$  中的较大者。
- (2) 语句 “if( $x < y$ )  $z = y$  else  $z = x$ ;” 是对  $x$ 、 $y$  作比较, 选择较大者赋值给  $z$ 。
- (3) 语句 “return  $z$ ;” 将  $z$  的值作为函数 max 的函数值 (即返回值)。返回值是通过函数名 max 带回到函数的调用处。
- (4) 在 main 函数中通过语句 “ $d = \max(a, b)$ ;” 调用函数 max, 求出  $a$ 、 $b$  中较大者, 并将它赋给变量  $d$ , 然后通过语句 “ $e = \max(d, c)$ ;” 再一次调用函数 max, 求出  $d$ 、 $c$  中的较大者, 并赋给变量  $e$ ,  $e$  即为 3 个数的最大值。

## 1.4.2 C 程序结构

C 程序由函数、编译预处理命令及注释 3 部分组成。

### 1. 函数

C 程序是由函数 (function) 构成的。一个完整的 C 程序可以由一个或多个函数组成, 其中必须有一个 main 函数 (主函数), 且只有一个 main 函数。C 程序执行时, 总是从 main 函数开始执行, 且与 main 函数在整个程序中的位置无关。

函数包括两部分。

(1) 函数首部, 即函数的第一行。包括函数名、函数类型、函数参数 (形参) 名、参数类型。例如, 例 1.3 中的 max 函数的首部为:

float	max	(float	x,	float	y)
↓	↓	↓	↓	↓	↓
函数类型	函数名	参数类型	参数名	参数类型	参数名

一个函数名后面必须有一对圆括号, 函数参数可以没有, 如 main()。

(2) 函数体, 即函数首部下面的花括号内的部分。

函数的结构形式如下:

```
函数类型 函数名(参数)
{ 函数体 }
```

函数体由各类语句组成, 执行时按语句的先后次序依次执行, 每个语句必须用分号 (;) 结尾。

### 2. 编译预处理命令

程序中每一个以 “#” 号开头的行, 是编译预处理命令, 一般放在程序的最前面。不同的编译预处理命令完成不同的功能。如 “#include<stdio.h>” 命令的作用是将 stdio.h 文件的内容嵌入到源程序中。

### 3. 注释

程序中的注释 (comment) 部分在编译时均被忽略掉, 所以注释在程序执行时不起任何