



高等院校计算机系列规划教材

C语言程序设计

刘明才 牟连泳 辛慧杰 编著



机械工业出版社
CHINA MACHINE PRESS



免费提供电子教案

下载网址 <http://www.cmpedu.com>



九江学院图书馆



1348860

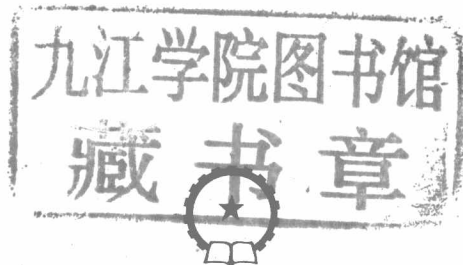
1359971

高等院校计算机系列规划教材

C 语言程序设计

刘明才 牟连泳 辛慧杰 编著

TP312/18237



机械工业出版社



本书主要介绍了 C 语言程序设计基础知识、各种数据类型和常用库函数、各种运算符和表达式、程序控制语句、数组、函数、编译预处理、指针、结构体与共用体、文件及 C 程序设计举例等。

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,并具备一定的程序设计能力。

本书按基础课程要求编写,适合作为大学各专业公共课教材和全国计算机等级考试参考书。

图书在版编目 (CIP) 数据

C 语言程序设计/刘明才,牟连泳,辛慧杰编著. —北京:机械工业出版社,2009.12

(高等院校计算机系列规划教材)

ISBN 978-7-111-29037-7

I. C… II. ①刘… ②牟… ③辛… 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 (2009) 第 206236 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:赵 轩

责任印制:李 妍

北京汇林印务有限公司印刷

2010 年 1 月第 1 版·第 1 次印刷

184mm×260mm·16 印张·390 千字

0001-3000 册

标准书号:ISBN 978-7-111-29037-7

定价:26.00 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

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者服务部:(010)68993821

封面无防伪标均为盗版

前 言

C 语言是一种非常优秀的程序设计语言，它既具备高级语言的特点，又具有直接操纵计算机硬件的能力，并因其丰富灵活的控制和数据结构、简洁而高效的语句表达、清晰的程序结构和良好的可移植性而拥有大量的使用者。目前，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，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。在本书的编写过程中，力求使其体现如下特点：

1. 对 C 语言中的重点、难点进行分解，将重点、难点分散编排，使学生的学习过程中循序渐进、平稳过渡。
2. 对 C 语言中生僻、不常用的内容不进行过多的描述，而对于在实践中使用较多、需要牢固掌握的内容，则进行了详细的叙述，并给出了大量的例子。
3. 本书在介绍 C 语言的基本知识的同时，还强调读者编程风格的形成，有意识地训练读者逐步养成良好的程序书写习惯和程序设计风格。
4. 本书所有例子均在 Turbo C 2.0 环境下调试通过，也能在 Visual C++ 6.0 环境下运行。

为了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 程序的基本构成与 Turbo C 的简单使用方法；第 2 章介绍了基本数据类型、常用输入与输出函数；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、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方法；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 章介绍了一些常用算法和程序设计举例。其中，第 1~5 章由刘明才编写，第 6~8 章由牟连泳编写，第 9~11 章由辛慧杰编写。

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，书中难免存在错误与不足，恳请读者批评指正。

作 者

目 录

前言

第1章 C语言概述	1
1.1 计算机语言	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程序例子	4
1.4.2 C程序结构	6
1.5 C程序在计算机上执行的步骤	7
1.5.1 C程序执行的一般步骤	7
1.5.2 用Turbo C 2.0运行C程序的步骤	8
1.6 本章小结	10
习题1	10
第2章 基本数据类型与常用库函数	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	13
2.3 常量与变量	16
2.3.1 常量	16
2.3.2 符号常量	18
2.3.3 变量	18
2.4 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	23
2.5 常用库函数	25
2.6 本章小结	26
习题2	26
第3章 运算符和表达式	29
3.1 算术运算符和算术表达式	29

3.2	自增、自减运算	31
3.3	位运算	32
3.4	赋值运算	35
3.5	关系运算与逻辑运算	37
3.5.1	关系运算	37
3.5.2	逻辑运算	38
3.6	条件表达式与逗号表达式	40
3.6.1	条件表达式	40
3.6.2	逗号表达式	41
3.7	本章小结	41
	习题3	42
第4章	C 语言的控制结构	45
4.1	算法与程序	45
4.1.1	算法	45
4.1.2	程序	45
4.2	顺序结构	47
4.3	选择结构	47
4.3.1	if 语句	47
4.3.2	switch 语句	54
4.4	循环结构	58
4.4.1	while 语句	58
4.4.2	do-while 语句	59
4.4.3	for 语句	60
4.4.4	break 语句	62
4.4.5	continue 语句	63
4.4.6	单重循环程序举例	63
4.4.7	多重循环结构	68
4.4.8	goto 语句	70
4.5	本章小结	71
	习题4	71
第5章	数组	76
5.1	一维数组	76
5.1.1	一维数组的定义	76
5.1.2	一维数组的存储形式	77
5.1.3	一维数组的初始化	77
5.1.4	一维数组的引用	78
5.1.5	一维数组的应用举例	78
5.2	二维数组	81

5.2.1 二维数组的定义	81
5.2.2 二维数组的存储形式	82
5.2.3 二维数组的初始化	82
5.2.4 二维数组的引用	83
5.2.5 二维数组应用举例	83
5.3 字符数组	85
5.3.1 字符数组的定义	85
5.3.2 字符数组的初始化	86
5.3.3 字符数组的输入/输出	86
5.3.4 常用字符串处理函数	87
5.3.5 二维字符数组	90
5.3.6 字符串应用举例	91
5.4 本章小结	93
习题5	93
第6章 函数	98
6.1 函数定义和函数调用	98
6.1.1 函数定义	98
6.1.2 return 语句	100
6.1.3 函数调用	100
6.1.4 函数声明	101
6.1.5 函数间的参数传递	103
6.2 函数的嵌套调用和递归调用	105
6.2.1 函数的嵌套调用	105
6.2.2 函数的递归调用	106
6.3 局部变量和全局变量	110
6.3.1 局部变量	110
6.3.2 全局变量	110
6.3.3 如何用 Turbo C 运行多文件的程序	114
6.4 变量的存储类别	114
6.4.1 auto (自动) 变量	115
6.4.2 static (静态) 变量	115
6.4.3 register (寄存器) 变量	116
6.5 内部函数和外部函数	116
6.6 本章小结	117
习题6	118
第7章 编译预处理	122
7.1 宏定义	122
7.1.1 不带参数的宏定义	122

7.1.2	带参数的宏定义	123
7.1.3	终止宏定义	125
7.2	文件包含	125
7.3	条件编译	126
7.4	本章小结	127
	习题7	128
第8章	指针	131
8.1	指针概述	131
8.1.1	指针的概念	131
8.1.2	指针变量	131
8.1.3	指针运算	134
8.1.4	指针作为函数的参数	135
8.2	指针与一维数组	138
8.2.1	一维数组元素的表示	138
8.2.2	数组名作为函数的参数	140
8.3	指针与二维数组	143
8.3.1	二维数组中的指针	143
8.3.2	指向二维数组元素的指针变量	145
8.3.3	二维数组的行指针作函数参数	146
8.4	指针与字符串	148
8.4.1	字符串的表示形式	148
8.4.2	字符串指针作函数参数	150
8.5	指针与函数	151
8.5.1	指向函数的指针	151
8.5.2	返回指针的函数	152
8.6	指针数组与多级指针	153
8.6.1	指针数组的概念	153
8.6.2	多级指针的概念	154
8.6.3	指针数组作 main 函数的形参	155
8.7	本章小结	156
	习题8	156
第9章	结构体与共用体	161
9.1	结构体类型的定义	161
9.2	结构体类型变量	162
9.2.1	结构体类型变量的定义	162
9.2.2	结构体变量的引用	164
9.3	结构体数组	165
9.3.1	结构体数组的定义与初始化	165

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 结构体做函数参数	168
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
9.10 本章小结	187
习题9	188
第10章 文件	192
10.1 文件的基本概念	192
10.2 文件的打开与关闭	192
10.2.1 文件类型指针	192
10.2.2 文件的打开	193
10.2.3 文件的关闭	194
10.3 文件的读写	194
10.3.1 字符读写函数 fgetc 和 fputc	195
10.3.2 字符串读写函数 fgets 和 fputs	197
10.3.3 数据块读写函数 fread 和 fwrite	198
10.3.4 格式化读写函数 fscanf 和 fprintf	199
10.4 文件的定位与随机读写	200
10.4.1 文件定位	201
10.4.2 文件的随机读写	201
10.5 本章小结	202
习题10	203
第11章 C 程序设计举例	206
11.1 常用算法	206
11.1.1 二分法	206
11.1.2 迭代法	207
11.1.3 枚举法	208
11.1.4 递推法	209
11.1.5 递归法	210
11.1.6 回溯法	212

11.2 程序设计实例	213
11.2.1 随机数的使用	213
11.2.2 日期和时间函数的使用	215
11.2.3 数值计算	219
11.2.4 学生成绩管理程序	223
附录	236
附录 A C 语言中的关键字	236
附录 B 字符与 ASCII 码对照表	237
附录 C 运算符的优先级与结合性	237
附录 D C 语言中的常用库函数	238
参考文献	243

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，也可以用来编写应用软件。

1.1 计算机语言

计算机语言 (computer language) 是指计算机系统能接受、理解和执行的语言。人们要用计算机完成某种数据处理任务，就需要把解决该方法 (步骤) 用计算机语言描述出来，即编写程序。程序就是计算机为完成某一项任务所必须执行的一系列指令。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。其中，机器语言和汇编语言也称为低级语言。

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)，它能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。每种型号计算机的机器指令，都有自己的编码格式，所组成的指令集合，则称为计算机的指令系统。计算机指令系统一般由数百条指令组成，每条指令都能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。

【例 1-1】 下面这段程序是使用 80x86 计算机的机器语言编写的，其功能是计算 $1+1$ 。

```
10111000
00000001
00000000
00000101
00000001
00000000
```

例 1-1 所示的这段程序，计算机可以直接执行。但对于人来说，就像“天书”一样，很难掌握。

2. 汇编语言

汇编语言 (assembly language) 是在机器语言基础上，引入助记符、符号数来表示机器指令中的操作码和操作数地址。助记符通常使用英文单词的缩写形式，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)。

【例 1-2】 用汇编语言计算 $1+1$ 的程序。

```
MOV AX, 1
```

ADD AX, 1

对于人来说, 这个用汇编语言编写的源程序, 比机器语言容易掌握。但是, 计算机是不能直接执行的。汇编语言源程序需要通过一个称为汇编程序 (assembler) 的软件, 将其翻译成计算机可以直接识别运行的机器语言程序, 称为目标程序 (object program)。例 1-2 所示的源程序通过汇编程序可以翻译成例 1-1 所示的目标程序。

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。计算机不能直接执行高级语言源程序, 因此需要一种称为编译程序的软件, 将其翻译成计算机可以直接识别运行的机器语言程序。

【例 1-3】 用 BASIC 语言计算并显示 $1+1$ 的程序。

PRINT 1 + 1

这个例子使用了 BASIC 语言中的输出语句 PRINT 来计算, 并在屏幕上输出 $1+1$ 的值。BASIC 语言是一种高级语言。目前, 高级语言已经发展到上千种, 常用的有 BASIC、Visual Basic、Pascal、Delphi、FORTRAN、COBOL、C、C++ 和 Java 等。其中, 影响最大、使用最广、寿命最长的是 C 语言。下面简要介绍 C 语言的发展历史。

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, 在一种称为 BCPL (basic combined programming language) 语言的基础上开发了一种新的语言, 称为 B 语言 (取 BCPL 的第一个字母)。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, Dennis M. Ritchie 在 B 语言的基础上设计出了 C 语言 (取 BCPL 的第二个字母)。1973 年, Ken Thompson 和 Dennis M. Ritchie 合作把原来用汇编语言编写的 UNIX 操作系统的 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 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) 合作出版了 “The C Programming Language” (《C 程序设计语言》)。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 有了很大的改进。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。新的标准被不断开发,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,并增加了一些数据库函数和开发了一些专用的、高度创新的特性,如可变长度的数组和 restrict 指针修饰符。这些改进再一次把 C 语言推到了计算机语言开发的前沿。但是 C99 的修改保持了 C 语言的本质特性,它仍然是一种简短、清楚、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,既可用于编写系统软件,也可用于编写应用软件。和其他高级语言比较,C 语言有如下主要特点。

(1) 简洁、紧凑、灵活

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) 中级语言

C 语言既有汇编语言的大部分功能,例如允许直接访问物理地址、进行位运算、直接对硬件访问等;也有高级语言的面向用户、容易记忆、容易学习和易于书写等特点。

(3) 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。它使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) 数据类型丰富

C 语言具有现代化语言的各种数据类型,并且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) 运算符丰富

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

(6) 移植性好

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,从而使 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,没有对程序员施加过多的限制,程序设计能够很自由、灵活性的进行,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,因此会降低可读性。

前面介绍的 C 语言的主要特点,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 程序例子

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

【例 1-4】 在屏幕上输出一行信息：

```
Hello, Everyone!
```

程序如下：

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

程序运行后，输出如下一行信息：

```
Hello, Everyone!
```

程序说明：

1) 第 1 行是预处理命令，它的作用是将文件 `stdio. h` 的内容嵌入到程序中，使输入输出能正常执行。每个预处理命令占一行，并以 `#` 号开头。有关预处理命令的详细内容将在第 7 章介绍。

2) 第 2 行是主函数的首部，主函数名为 `main`，前面的 `void` 表示函数返回值的类型，`void` 的含义是空，表示函数不返回任何值。函数名后面必须有一对圆括号。主函数也称为 `main` 函数。

3) 第 3~5 行是由大括号括起来的函数体。函数体由若干语句组成，用于完成函数的功能。本例中的函数体只有一个语句，是由库函数（系统已定义的函数）`printf` 构成的。`printf` 函数的功能是将其后圆括号中双引号内的字符串原样输出，“`\n`”是换行符，其作用是输出一个换行，使后面的内容在下一行输出。有关 `printf` 函数的一般用法将在第 2 章介绍。

4) C 语言规定每个语句以分号结尾，分号是语句不可缺少的部分。第 1 行的预处理命令不是 C 语句，它在编译之前执行，因此末尾不加分号。

【例 1-5】 一个简单的加法计算程序，求两个整数之和并输出。

程序如下：

```
#include < stdio. h >          /* 预处理命令 */
void main( )                  /* 主函数首部,下面是函数体 */
{
    int x,y,sum;              /* 定义存放整数的变量 x,y,sum */
```



```
printf("Input two integers:"); /* 输出信息"Input two integers:" */
scanf("%d%d",&x,&y); /* 给变量 x,y 输入整数 */
sum = x + y; /* 计算 x+y, 把结果赋给 sum */
printf("sum = %d\n",sum); /* 输出 sum 的值 */
}
```

程序运行情况如下:

```
Input two integers:3 5 ↵
sum = 8
```

程序说明:

1) /* */ 是注释部分, 在程序执行中不起任何作用, 是程序编写者写给程序阅读者的一些说明 (注释), 以帮助阅读者理解程序。任何文字都可用 “/*” 和 “*/” 括起来, 作为程序的注释部分。注意, 在 “/” 和 “*” 之间不能有空格。注释部分可以出现在程序的任何地方。

2) main 函数的函数体由 5 个语句组成。第 1 个语句 “int x, y, sum;” 是变量定义语句, 用于定义存放整数的变量 x、y 和 sum。变量是用来存放数据的, 它代表内存中的存储单元。int 是 integer (整数) 的缩写, 是一个关键字, 用来定义整型变量等。C 语言中的关键字也称为保留字, 在 C 语言中有特殊的作用, 不能用来作为其他对象的名字。关键字共 32 个, 见附录 A。

3) 语句 “scanf (“%d%d”, &x, &y);” 是给变量 x 和 y 赋值。scanf 是库函数中的一个输入函数, “%d” 是格式说明符, 表示输入一个整数, “&x” 表示 x 的地址 (即变量 x 所代表的存储单元的地址)。该语句表示输入两个整数, 分别赋给 x 和 y。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数据。在本书中, 用符号 “↵” 表示回车 (〈Enter〉) 键, 如 “3 5 ↵” 表示从键盘输入 3、空格 (将两个数据分开)、5, 再按回车键。

4) 语句 “sum = x + y;” 是赋值语句, 用于计算变量 x 与 y 之和, 并把结果赋给变量 sum。

5) 语句 “printf (“sum = %d\n”, sum);” 中的 “%d” 是格式说明符, 表示在该位置输出一个整数, 即 sum 的值, 如在本例中输出 “sum = 8”。

【例 1-6】 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) /* 定义求两个实数较大值的函数 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; /* 定义存放实数的变量 a,b,c,d */
  printf("a,b,c=?\n"); /* 输出信息"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 max 函数,将 a,b 中较大者赋给 d */
d = max(d,c);          /* 调用 max 函数,将 d,c 中较大者赋给 d */
printf("max = %f\n",d); /* "%f"表示在此位置输出变量 d 的值 */
}

```

程序运行情况如下:

```

a, b, c = ?
3 8 -6 ✓
max = 8.000000

```

程序说明:

1) 程序第 2 行是一个用户自定义函数,其功能是求两个实数 x 与 y 中的较大者。函数名为 `max`,函数值类型为 `float`,表示实数。`float` 也是一个关键字,可用于定义实型变量等。对于一般函数的定义方法和调用方法在第 6 章介绍。

2) 语句 “`if (x < y) z = y; else z = x;`” 是一个选择语句,表示当 $x < y$ 时,执行 $z = y$; 否则执行 $z = x$; 即该语句的功能是对 x 、 y 作比较,选择较大者赋给 z 。选择语句将在第 4 章介绍。

3) 语句 “`return z;`” 将 z 的值作为函数 `max` 的函数值 (即返回值)。

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, 并把它赋给变量 d , 然后通过语句 “`d = 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 d , 此时变量 d 已是新的值 (原先的值已消失), 即 d 为 3 个数中的最大值。

5) 语句 “`printf ("max = %f\n", d);`” 中的 “`%f`” 是格式说明符, 表示在该位置输出一个实数值, 即 d 的值, 如在本例中输出: `max = 8.000000`, 其中小数部分输出 6 位。

1.4.2 C 程序结构

C 程序是由函数 (function) 构成的, 函数是 C 程序的基本单位。

1) 一个 C 程序必须有一个 `main` 函数 (主函数), 且只有一个 `main` 函数。C 程序总是从 `main` 函数开始执行, 且与 `main` 函数在程序中的位置无关。`main` 函数执行完毕, 整个程序的执行就结束。

2) 根据需要, 一个 C 程序可以不包含用户自定义函数, 也可以包含多个用户自定义函数。例如, 在例 1-6 中定义了一个用户自定义函数 `max`。`main` 函数调用这个 `max` 函数求两个数的最大值。

函数包括两部分:

① 函数首部, 即函数的第 1 行。它包括函数名、函数类型、函数参数 (形参) 名、参数类型。例如, 例 1-6 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, 但函数可以没有参数。例如, 前面例子中对主函数

main 的定义。

② 函数体，即函数首部下面的大括号 {……} 内的部分。函数体由各类语句组成，并按语句的先后顺序依次执行，每个语句必须以分号 (;) 结尾。

3) 在函数中可以调用系统提供的库函数，在调用之前只要将相应的头文件 (header files) 包含到程序中即可。例如，当使用系统提供的输入/输出函数时，需要用预处理命令 (preprocessor directive)：

```
#include <stdio.h>
```

将头文件 stdio.h 包含到程序中。

1.5 C 程序在计算机上执行的步骤

1.5.1 C 程序执行的一般步骤

用 C 语言编写的源程序不能立即在计算机上执行，必须由编译程序将其翻译成用机器语言 (由 0, 1 组成的二进制指令) 构成的目标程序，然后将目标程序与系统的函数库中有关的函数 (由二进制指令构成) 连接起来，形成可执行程序。C 语言的编译系统主要由编译程序 (compiler)、连接程序 (linker)、函数库等组成。如果要使 C 程序在一台计算机上执行，必须经过编辑、编译和连接，最后得到可执行程序 (executable program)。

(1) 编辑

编辑是建立或修改 C 源程序文件的过程，并以文本文件的形式存储在磁盘上，其扩展名为 .c。

(2) 编译

C 语言是计算机高级语言，由它编写的源程序必须经过编译程序对其进行编译，生成目标程序文件，其扩展名为 .obj。

(3) 连接

机器可以识别编译生成的目标程序，但不能直接执行，还需要将目标程序与库函数进行连接处理，连接工作由连接程序完成。经过连接后，生成可执行程序文件，其扩展名为 .exe。

(4) 运行

C 程序经过编译、连接后生成了可执行文件。要运行可执行文件，既可在编译系统环境下运行，又可脱离编译系统，直接在 Windows 资源管理器下双击可执行文件名运行。C 程序运行的全过程如图 1-1 所示。

由图 1-1 可以看出，当编译或连接出现错误时，说明 C 程序编写时有语法错误；若在运行时出现错误或结果不正确，则说明程序设计有问题 (逻辑错误)，都需要修改源程序并重新编译、连接和运行，直到程序运行正确为止。最后得到可执行文件 f.exe。该文件可以脱离 C 编译系统，直接在计算机上运行。例如，在输入例 1-5 的源程序时，漏写语句中的任何一个分号、逗号、括号都会导致语法错误，从而编译时出错。如果将程序中的语句 “sum = x + y;” 错写成 “sum = x - y;” 将导致程序执行时的结果不正确，说明程序有逻辑错