



国家开放大学
THE OPEN UNIVERSITY OF CHINA

C语言程序设计

(第2版)

任爱华 主编



中央广播电视大学出版社



国家开放大学
THE OPEN UNIVERSITY OF CHINA

C LANGUAGE
PROGRAMMING

C 语言程序设计

(第2版)

任爱华 主编

中央广播电视大学出版社·北京

图书在版编目 (CIP) 数据

C 语言程序设计 / 任爱华主编. —2 版. —北京: 中央广播电视大学出版社, 2014. 12

ISBN 978-7-304-06900-1

I. ①C… II. ①任… III. ①C 语言-程序设计-开放大学-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 002188 号

版权所有, 翻印必究。

C 语言程序设计 (第 2 版)

C YUYAN CHENGXU SHEJI

任爱华 主 编

出版·发行: 中央广播电视大学出版社

电话: 营销中心 010-66490011 总编室 010-68182524

网址: <http://www.crtvup.com.cn>

地址: 北京市海淀区西四环中路 45 号 邮编: 100039

经销: 新华书店北京发行所

策划编辑: 邹伯夏

版式设计: 赵 洋

责任编辑: 吴国艳

责任校对: 宋亦芳

责任印制: 赵连生

印刷: 北京密云胶印厂

印数: 0001 ~ 3000

版本: 2015 年 1 月第 2 版

2015 年 1 月第 1 次印刷

开本: 787 × 1092 1/16

印张: 20.5 字数: 455 千字

书号: ISBN 978-7-304-06900-1

定价: 35.00 元

(如有缺页或倒装, 本社负责退换)

中央广播电视大学经过三十多年的努力与发展,已经更名为国家开放大学。本书是为国家开放大学计算机科学与技术等专业开设的“C 语言程序设计”课程而编写的一本教材。

同第1版相比,第2版的内容更加丰富,使用更为方便。在第4章“4.5 数组的应用”中,增加了二分查找、插入排序等实用内容;在第7章增加了链表的内容,能够为学习后续的数据结构课程打下更好的基础。这次修订的重点在于丰富附录的内容,一是增加了练习题参考解答,二是增加了在输入输出函数调用中使用的格式字符串的详细说明,三是增加了C语言系统函数库中常用函数的列表及说明,这些内容都为自学者提供了方便,适合在职和业余者学习。

第2版按照教学大纲要求对全书内容进行了系统和全面地梳理、充实和提高,文字叙述更加通顺,例题分析更加透彻,程序编写更加规范,章节结构和内容取舍更加合理,练习题量得到进一步充实和丰富。

本书共分为8章,第1章为C语言概述,介绍了C语言中的字符集、单词、语句、函数、程序的概念,以及相互之间的关系,同时介绍了键盘输入和屏幕输出数据的方法,以及在VC++ 6.0 软件开发环境下,上机输入、编辑、编译、连接、运行程序的全过程。第2章为基本数据类型与表达式,介绍了C语言中各种基本数据类型,以及常量、变量、函数、表达式的表示和求值方法。第3章为流程控制语句,介绍了各种分支、循环、跳转等流程控制语句的语法规则和在程序设计中的具体应用。第4章为数组和字符串,介绍了它们的定义和在数值计算及数据处理方面的具体应用。第5章为指针,介绍了指针的定义和使用、指针的各种运算、指针与数组的关系、变量存储空间的动态存储分配等内容。第6章为函数,介绍了用户函数的定义、声明、调用的格式和方法,变量的不同作用域,以及递归函数和函数指针的概念。第7章为结构与联合,介绍了结构的定义,结构变量的定义和结构成员的访问方法,结构与链表的构成及相互关系,联合的定义与使用。第8章为文件,介绍了文件和文件流的概念,数据文件的打开与关闭,对文本文件和二进制文件的访问操作。

书中带*章节为不考核内容,学生可根据自身的情况进行选择学习。

书后提供了练习题参考解答。同学们一定要独立思考认真做练习,只有这样才能够灵活地运用知识,巩固和提高所学知识,增强分析问题和解决问题的能力。参考解答起到自查练习效果的作用。

本书本版次仍由北京航空航天大学任爱华教授主编和统稿，由国家开放大学徐孝凯教授和北京开放大学张纪勇教授编写前后各4章初稿。

由于作者水平有限，错误和不足之处在所难免，敬请专家和读者批评指正。

编 者
2014 年 10 月

C 语言程序设计课程是计算机专业的入门课程。C 语言是一种最早出现和流行的计算机高级程序设计语言，经久不衰，它具有丰富的数据类型和各种运算功能，带有庞大的函数库，支持过程化和结构化的程序设计，是系统软件和应用软件开发人员的主要开发工具之一，是学习计算机专业其他课程的基础。

C 语言内容丰富，本书介绍了每一种标准数据类型的含义，每一种运算符的表示，各种表达式和语句的构成，枚举、数组、结构、指针等数据类型的定义和使用，函数的定义、声明与调用格式，结构化和模块化程序设计方法等内容。

对于书中介绍的每个知识点，不仅从概念叙述上尽量做到条理清楚和层次分明，而且通过典型性和针对性强的程序分析和编程举例加以解释和阐述，达到学以致用、举一反三之目的。

本书共分为 8 章，依次为 C 语言概述、基本数据类型和表达式、流程控制语句、数组和字符串、指针、函数、结构与联合、文件等。各章之间的内容连贯有序，衔接自然，成为一个有机的整体。

C 语言课程是一门实践性极强的课程，只有多阅读和分析已有的程序，多练习编写程序，多上机调试和运行程序，才能够获得真正的知识。书中所有例题和习题的程序均可以作为上机题使用。

书中每一章后面都给出了具有各种题型的练习题和实验题。上机实验要至少安排 6 次，从全书 8 章后面的上机实验题中任选，每次上机时间要不低于 3 个机时。

本书中所有程序都在 Microsoft Visual C++ 6.0 开发环境下调试运行通过，避免出现语法错误和逻辑错误。

本书由北京航空航天大学任爱华教授主编和统稿，由中央电大徐孝凯教授和北京电大张纪勇副教授分别提供前、后各 4 章内容的初稿。由于作者水平所限，错误和不足之处在所难免，敬请专家和读者批评指正。

在本书编写和审定过程中，得到了首都经贸大学李宁、北京航空航天大学李书通，北京工商大学刘杰，海南大学钟声、中央广播电视大学任为民和刘臣等专家和教授的真诚帮助和指正，就此表示衷心感谢。

编者

2008 年 3 月

第1章 C 语言概述

1

- 1.1 引言 1
- 1.2 C 语言字符集 2
- 1.3 C 语言单词 4
- 1.4 C 语句 7
- 1.5 C 函数 9
- 1.6 C 程序 13
- 1.7 printf() 和 scanf() 函数介绍 15
- 1.8 VC++ 6.0 集成开发环境简介 19

第2章 基本数据类型与表达式

33

- 2.1 数据类型 33
- 2.2 常量 36
- 2.3 变量 41
- 2.4 运算符和表达式 46
- 2.5 函数 55

第3章 流程控制语句

66

- 3.1 流程控制语句概述 66
- 3.2 if 语句 67
- 3.3 switch 语句 71
- 3.4 for 语句 76
- 3.5 while 语句 86
- 3.6 do...while 语句 90
- 3.7 跳转类语句 96

第4章 数组和字符串 108

- 4.1 数组的概念 108
- 4.2 一维数组 109
- 4.3 二维数组 114
- 4.4 使用 typedef 语句定义数组类型 119
- 4.5 数组的应用 123
- 4.6 字符串 136

第5章 指针 152

- 5.1 指针的概念 152
- 5.2 指针变量 153
- 5.3 指针运算 160
- 5.4 指针与数组 164
- 5.5 动态存储分配 168

第6章 函数 178

- 6.1 函数定义 178
- 6.2 函数调用 181
- 6.3 变量作用域 187
- 6.4 递归函数 193
- * 6.5 函数指针 197

第7章 结构与联合 206

- 7.1 结构的定义 206
- 7.2 结构变量的定义和初始化 210
- 7.3 结构成员的访问 214
- 7.4 使用结构的程序举例 214
- 7.5 结构与函数 219
- 7.6 结构与链表 223
- 7.7 联合 227

第8章 文件 239

- 8.1 文件的概念 239
- 8.2 数据文件的打开和关闭 241

- 8.3 文本文件的访问操作 244
- 8.4 二进制文件的访问操作 247

附录 259

- ASCII 代码表 259
- 格式字符串的有关说明 261
- C 语言系统函数库中的部分函数简介 268

练习题参考解答 302

参考文献 317

本章主要介绍 C 语言中所包含的字符、单词、语句等基本语法成分的定义及作用，C 语言函数和程序的构成，以及在 Visual C++ 6.0 集成开发环境下的 C 语言程序的上机操作过程。通过本章学习，要求达到：

- 了解 C 语言的基本语法规则和程序结构；
- 了解 Visual C++ 6.0 集成开发环境窗口的布局和各部分的功能；
- 了解函数原型、函数定义和函数调用的各自作用；
- 掌握 printf() 函数和 scanf() 函数的调用格式和使用。

1.1 引言

C 语言是使用时间最久和最普及的计算机高级程序设计语言之一，是人们学习计算机程序设计的首选语言，也是学习其他计算机课程和进行软件开发的基础。

C 语言从 20 世纪 60 年代问世以来，长盛不衰，它属于面向过程的程序设计语言，兼有汇编语言和高级语言的双重特点，一直被计算机系统软件开发人员所喜爱和使用。在 C 语言的基础上经过不断地优化、扩展和改版，于 20 世纪 80 年代以后发展成为既面向过程又面向对象的 C++ 语言，以适应软件开发技术从面向过程转向面向对象的迫切要求。但 C++ 语言的基础仍然是 C 语言，学好 C 语言，将为以后进一步学习面向对象的 C++、C# 或 Java 等语言奠定基础。

面向过程是求解问题的一种传统方法，它采用自顶向下、逐步求精的分析过程，把待解决的整个问题按功能划分为若干个相对独立的小问题，每个小问题又可以按功能划分为若干个相对独立的更小问题，依此类推，直到最低一层的问题较容易用一种计算机语言编写的程序模块实现为止。在面向过程的程序设计中，每个程序模块具有一定的相对独立的功能，通过较小的程序功能模块的调用组合就可以形成较大的程序功能模块，最后形成一个解决整个问题的完整程序。在采用面向过程的方法进行程序设计时，整个程序的功能是通过程序模块之间的相互调用完成的，若问题比较复杂，程序结构即模块之间的调用关系也会变得复杂和混乱，并且也容易增加模块之间的依赖性以及调试和修改程序的难度。

面向对象是求解问题的一种新的思路和方法, 它把求解问题中的所有事物 (即独立个体) 都作为各自不同的对象, 进而把具有共同特征的对象归属为一个类, 由此得到若干个不同的类, 每个类是对该类事物 (对象) 的抽象描述, 通过相同或不同类对象之间的相互作用和通信使问题得以解决。

C 语言有标准版本, 但各软件公司开发的 C 语言版本并不严格遵守它, 而是与它兼容且稍有修改和扩充。现在普遍使用的 C 语言版本有 Turbo/Borland、Microsoft 等公司开发的产品, 并且针对 DOS、Windows、UNIX 等操作系统的不同平台有不同版本。本书所采用的 C 语言版本将包含在 Microsoft 公司开发的 Visual C++ 6.0 (Microsoft Visual C++ 6.0 的缩写) 版本之中。在此版本的运行环境中, 若程序文件扩展名指定为 c, 而不是缺省时系统自动添加的 cpp, 则将使用其中的 C 语言运行环境, 编译和运行 C 语言程序; 若程序文件扩展名指定为 cpp, 则自动使用 C++ 语言运行环境, 编译和运行 C++ 语言程序。由于 C++ 语言完全包含 C 语言, 或者说 C 语言是 C++ 语言中的一个真子集, 所以 C 语言程序完全能够在 C++ 语言运行环境下编译和运行, 反之则不然。本书以 Visual C++ 6.0 为依据, 向读者介绍 C 语言的基本内容和进行一般程序设计的知识, 使读者能够根据实际问题设计出 C 语言应用程序并能够上机运行得到正确结果。

计算机程序设计语言是人们同计算机进行交流的工具, 其语法结构同人类使用的自然语言 (如汉语、英语等) 类似, 也具有字、词、句、章等的层次结构和相应的构成规则。例如, 对于 C 语言来说, 它具有规定的字符集, 由字符集中的一个或若干个字符按照相应的组词规则就可以构成合法的 C 语言单词, 由一个或若干个 C 语言单词按照相应的语法规则就可以构成 C 语言语句, 而完成某一功能的一条或若干条 C 语言语句通过适当包装就可以构成一个程序模块——函数, 由一个或若干个函数以及一些附加的成分就可以构成一个独立的 C 语言程序文件, 由一个或若干个程序文件就可以构成一个完整的 C 语言程序。一个 C 语言函数相当于一篇文章中的段落, 一个 C 语言程序文件相当于一篇文章, 一个完整的 C 语言程序则相当于一个文档, 它包含有多篇文章, 即多个程序文件。

在 C 语言中, 从最基本的字符成分开始, 通过逐级构造, 最后形成一个完整的程序, 此过程可形象化地表示如图 1-1 所示。

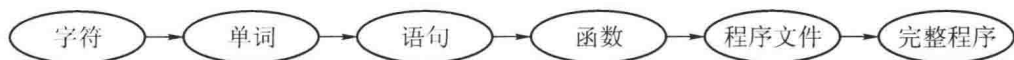


图 1-1 一个 C 语言程序的构造过程

1.2 C 语言字符集

下面分类列出在 C 语言中规定的全部字符, 除此之外的符号都是不允许在 C 语言程序使用的。

1. 大、小写英文字母 (52 个)

大写字母从 A 到 Z, 小写字母从 a 到 z, 共计 52 个。

在 C 语言中, 同一个英文字符的大写和小写被视为不同的字符, 此种特性被称为大小写敏感的, 这与其他计算机语言中的规定不同, 其他许多语言是大小写不敏感的, 一个字母的大写和小写被视为同一字符。在 C 语言中的标识符 Abc、abc、ABC 等是完全不同的标识符, 可以用来表示相应不同的对象。

2. 十进制数字符号 (10 个)

十进制数字符号 0, 1, 2, 3, ..., 9, 共计 10 个。

3. 标点符号 (10 个)

C 语言中的标点符号如表 1-1 所示。

表 1-1 C 语言中的标点符号

序号	符号	名 称	作 用
1	,	逗号	数据之间的分隔符
2	;	分号	简单语句结束符或 for 循环头中的表达式分隔符
3	'	单引号	字符常量起、止标记符
4	"	双引号	字符串常量起、止标记符
5	:	冒号	语句标号结束符或条件运算符
6	<space>	空格	语句中各单词成分之间的分隔符, 对应键盘上的空格键
7	{	左花括号	复合语句的开始标记符
8	}	右花括号	复合语句的结束标记符
9	<CR>	回车符	用 <CR> 表示按下键盘上的回车 (Enter) 键
10	<Tab>	制表符	对应键盘上的 Tab 键

4. 单字符运算符 (19 个)

C 语言中的单字符运算符如表 1-2 所示。

表 1-2 C 语言中的单字符运算符

序号	符号	名 称	作 用
1	(	左圆括号	同右圆括号配对使用, 用于表达式和函数运算
2	)	右圆括号	同左圆括号配对使用, 用于表达式和函数运算
3	[	左中括号	同右中括号配对使用, 用于数组元素访问
4	]	右中括号	同左中括号配对使用, 用于数组元素访问
5	+	加号或正号	用于加法运算或取正值

续表

序号	符号	名 称	作 用
6	-	减号或负号	用于减法运算或取负值
7	*	乘号或间接访问符	用于乘法运算或进行间接访问运算
8	/	除号	用于除法运算
9	%	取模运算符	用于取两个整数相除的余数
10	.	小数点或成员访问符	用于标记小数位置或指定结构数据成员
11	<	小于号或左尖括号	当作为尖括号时与右尖括号配对
12	=	赋值号	把一个值赋予一个变量
13	>	大于号或右尖括号	当作为尖括号时与左尖括号配对
14	!	感叹号	逻辑非运算符
15	~	波折号	按位取反运算符
16	&	和符号	取地址或按位与运算符
17	^	尖字符	按位异或运算符
18		竖线	按位或运算符
19	?	问号	条件表达式运算符

5. 特殊用途符号（3 个）

井字符：编译预处理命令行的开始标记符。

\ 反斜线：转义字符序列的开始标记符。

_ 下划线：通常用于保留字和标识符中。

6. 字符和字符串常量数据中所使用的字符

字符常量和字符串常量是分别使用单引号和双引号作为起止定界符作为标记的数据（单词），如'A'和"xyz"分别是字符常量和字符串常量。在字符常量中，能够使用 ASCII 代码表（见附录 B）中的任何字符。如@和\$都不属于 C 语言中的字符，但都可以使用在 C 语言字符常量中，分别表示为'@'和'\$'。在字符串常量中，能够使用从计算机输入得到的任何字符和符号。如Σ、Π、π等全角符号都不属于 C 语言中的字符，但都可以使用在 C 语言字符串常量中。每个汉字也都不属于 C 语言中所规定的字符，同样也能够使用在 C 语言字符串常量中，如"\$ 25.64"、"姓名"、"x±y≠z"等都是合法的 C 语言字符串。

1.3 C 语言单词

由 C 语言字符按照一定的组词规则可以构成各种不同用途的 C 语言单词，可以把 C 语言单词分为以下保留字、标识符、常量、运算符和分隔符五类。

1. 保留字

C 语言保留字又称关键字，它是 C 语言系统内部已经定义的、具有确定含义的一组英文单词。VC++ 6.0 中关于 C 语言部分的全部保留字如表 1-3 所示。

表 1-3 C 语言保留字表

保留字	保留字	保留字	保留字	保留字	保留字
auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	int	long
return	short	signed	sizeof	register	static
struct	switch	typedef	union	unsigned	void
volatile	while				

每个保留字都被 C 语言系统赋予了一定的含义，用作各种语句的框架，即关键字。例如，保留字 int 表示整数类型，是定义整数变量的语句关键字；保留字 return 含义为返回，是构成返回语句的关键字；等等。因此，不能在程序中使用保留字做其他用途，也就是说，用户定义的任何对象的名字不能与保留字相同。

另外，在以井字符开头的预处理命令中，其命令关键字虽然不被算做 C 语言保留字，但也最好把它们作为保留字看待，不要使用在其他地方，以免引起混乱。这些预处理命令关键字有 include、define、if、else、ifdef、ifndef、endif、undef 等。

在 VC++ 6.0 开发环境中，输入和显示一个 C 语言程序时，每个 C 语言保留字和预处理命令关键字均以醒目的蓝颜色字显示，注释内容以绿颜色字显示，其他语言成分中的字符均以黑颜色字显示出来。

2. 标识符

标识符是用户在程序设计中为所使用的对象（如变量、函数、自定义数据类型等）进行标识的名字。在 C 语言中规定：每个标识符必须是由英文字母、十进制数字符号和下划线排列而成的一串字符，并且第一个字符必须是英文字母或下划线。每个标识符中的字符数可以任意，但只有前 32 个字符有效。通常，一个标识符由 1~32 个字符组成。例如，a、ab、size、Max、x1、y25、fun_1、Student_Num 等都是合法的标识符；3xy、“work”、lable:、Hi-4、list length 等都是非法的标识符，因为它们中有的第一个标识符以数字开头，有的（后 4 个）标识符中使用了非法字符（最后一个含有空格）。

在编写 C 语言程序时，需要给每个待使用的对象命名一个标识符，为了便于记忆和阅读，最好使用该对象的英文单词（词组）或汉语拼音作为标识符，有时将第一个字母大写，有时使用下划线连接两个英文单词或拼音单词。例如，可以用 wages、Wage、Gongzi、gongZi 等表示工资，用 Name、xingMing、XM 等表示姓名，用 maxWage、MaxWage、max_wage 等表

示最高工资值。

3. 常量

常量分为数值常量、字符常量和字符串常量三类。日常使用的十进制常数可以直接作为 C 语言中的数值常量 (简称常数) 使用, 它又可以被细分为整数常量和实数常量 (带小数点常量) 两种。例如, 32、-128、+100、3.26、-50.718 等都是合法的 C 语言数值常量, 其中前 3 个为整数常量, 后 2 个为实数常量。整数常量简称为整数, 实数常量简称为实数。

C 语言字符常量就是单个 ASCII 码字符, 表示时为了把它同单个字符的标识符或数值相区别, 必须用单引号括起来。如 'a'、'B'、'+'、';'、'5' 等都是字符常量, 但单独写 5 则为常数, 单独写字母 a 则为标识符, 单独写加号 (+) 则为运算符, 单独写分号 (;) 则为标点符。使用以单引号括起来的转义字符也是一种字符常量, 使用它来表示回车、换行、单引号、双引号等字符常量, 有关这方面的内容将在第 2 章详细介绍。

字符串常量就是由 ASCII 码字符、汉字区位码字符以及能够由计算机输入的任何字符所组成的一串字符。同样, 为了把它同其他语法成分相区别, 表示时必须用双引号括起来。例如, "1234"、"a + b ="、"main:"、"x, y, z ="、"查找失败"、"1. 输出线性表的长度" 等都是字符串常量, 简称为字符串或串。

现在国际上统一采用 Unicode 字符编码体系, 它能够包含全世界所有国家和民族使用的字符, 每个字符用两个字节 (即二进制 16 位) 编码, 原来一直使用的 ASCII 码变为其中的一个子集, 该子集中高字节的 8 位全用 0 填补, 低字节的 8 位保存原来的 ASCII 码。严格地说, 一个字符串中的字符, 可以使用 Unicode 字符编码表中的任何字符。现在计算机中的所有软件环境 (如 Word、Excel、PowerPoint、VC++、Java 等), 都提供有 Unicode 字符编码表。

由以上讨论可知: a、'a' 和 "a" 是完全不同的语言成分, 它们分别对应为一个标识符、一个字符常量和一个字符串。

4. 运算符

运算符是对数据进行运算的符号。C 语言中的运算符有单字符运算符, 如 +、-、*、/ 等; 有双字符运算符, 如 <=、!=、->、++、&&、*= 等; 也有三字符运算符, 如 <<=、>>= 等。另外, 还有一个特殊运算符, 它是用 C 语言保留字 sizeof 表示的, 用来计算出一个数据所占用的存储字节数。以后有关章节将陆续解释每个 C 语言运算符的含义及作用。

由运算符和操作数可以构成各种表达式, 对表达式进行计算的结果将得到一个确定的值。例如, 25 * 4 + 6 就是一个数值表达式, 其求值结果为 106。当然一个单独的操作数也被看做一个表达式, 因为它本身就是一个值或对应一个值。如一个常数 25, 一个变量 x, 一个字符 '!', 一个字符串 "apple" 等操作数都是表达式, 它们为最简单的表达式, 其值能够直接使用而无须计算得到。

5. 分隔符

分隔符的作用就是分隔前后两个相邻的语法成分 (单词、语句、命令等), 或者是结束

前一个语法成分，开始后一个语法成分。能够作为分隔符的字符有逗号、分号、冒号、空格、制表符、回车符等。例如，“`int x=3, y=5;`”是一条变量定义语句，开始的保留字 `int` 和变量标识符 `x` 之间必须使用至少一个空格，将它们分隔开，其后的逗号起到分隔不同变量定义的作用，最后的分号起到结束此语句的作用。C 语言中不同的分隔符起到不同的分隔作用，这将在以后各章的语句格式中介绍。

1.4 C 语句

由 C 语言中的单词按照一定的语法规则排列起来就构成语句。虽然每种语句的语法规则不同，但除了复合语句外，最后都必须以分号结束。

【例 1-1】

```
(1) int x;  
(2) x=20*35-6;  
(3) if(x>0)scanf("%d",&x);  
(4) break;  
(5) typedef int DataType;  
(6) void Sort(int aa[],int mn);  
(7) Sort(a,n);  
(8) printf("x=%d\n",x);
```

其中，每一行都是一条 C 语句，当然其前面的编号除外。

在第 (1) 条语句中，含有 4 个单词，依次为表示整型的保留字 `int`、空格、表示变量的标识符 `x` 和最后的分号。在第 (2) 条语句中含有 8 个单词，依次为变量标识符 `x`、赋值号 `=`、常数 20、乘号 `*`、常数 35、减号 `-`、常数 6 和最后的分号。第 (3)~第 (8) 条语句中，分别含有 15、2、6、15、7 和 7 个单词。

按照语句功能，可以把 C 语句分为以下 8 类：

1. 类型定义语句

类型定义语句用来定义系统预定义（标准、内置）类型之外的，用户根据特定需要而使用的数据类型。如结构、联合、枚举类型都需要用户结合应用情况按照一定的语法规则给出具体定义。例如，“`struct Student {char name [10]; int age;};`”就是一条结构类型的定义语句，定义的结构类型名为 `Student`，它包含有姓名（`name`）和年龄（`age`）这两个数据成员。

2. 变量定义语句

变量定义语句用来定义程序中需要使用的属于某个类型的变量。如【例 1-1】中第 (1) 条语句定义了一个整型变量 `x`，其中 `int` 表示系统预定义的整数类型，`x` 为用户命名的

一个变量标识符,以后可以用它来表示(即保存)一个整数,该整数的物理含义由具体应用而定,如可以用来表示一个人的年龄、一件物品的数量、一个单位的人数等。在【例 1-1】中第(2)条语句中,用 `x` 保存一个整数 694,它是赋值号右边表达式 $20 * 35 - 6$ 的值。

在一条变量定义语句的前面加上 `const` 保留字则为常量定义语句,它是变量定义语句的一种特殊情况,以后将会详细介绍。

3. 函数原型语句

函数原型语句又称函数声明语句,用它来声明一个函数存在并给定调用格式。一般情况下,函数原型语句出现在一个程序文件的开始,即所有的函数定义模块之前,以便在其后的每个函数模块中知道如何调用一个函数。与函数原型语句对应的是函数定义模块,它可以出现在程序文件中的任何位置,甚至可以出现在同一个程序的其他程序文件中。例如,【例 1-1】中第(6)条语句就是一条函数原型语句,通过这条语句能够看出:用户调用该函数时需要使用两个实际参数,一个为整型数组,另一个为整型数,而且该函数执行后不返回任何值。

4. 表达式语句

在任何一个合法的 C 语言表达式后加上一个语句结束符分号就构成了一条语句,称此语句为表达式语句。最常用的表达式语句为赋值表达式语句和函数调用表达式语句。例如,【例 1-1】中第(2)条为赋值表达式语句,第(7)条和第(8)条均为函数调用表达式语句。

5. 复合语句

用花括号括起来的语句序列合起来称为一条语句,即复合语句。例如, `{int x = 3, y; y = x + 2;}` 就是一条复合语句,它包含有两条简单语句,前者为变量定义语句,后者为赋值表达式语句。在一条复合语句中可以包含任意多条语句(包括不含任何语句),并且每条语句可以是任何种类的语句,当然仍可以是复合语句。例如, `{}, {};`, `{x = 40;}` 等都是合法的复合语句,其中 `{}` 中不含有语句; `{};` 中含有一条空语句,即只有分号的语句被称为空语句,它也是一条合法的语句; `{x = 40;}` 中含有一条简单的赋值语句。

一条复合语句是以左花括号作为开始标记,以右花括号作为结束标记的,其后不需要使用分号;若误用分号,计算机系统则认为是后接了一条空语句。例如,“`{int a; a = 1;};`”为两条语句,前者为复合语句,后者为空语句。

复合语句同任何以分号结束的简单语句一样,能够被使用在允许语句出现的任何地方。在以后叙述中提到的语句既包括简单语句也包括复合语句。

6. 选择类语句

选择类语句能够按照条件表达式的不同取值选择执行不同的语句模块。例如,【例 1-1】中第(3)条语句就是一条条件选择语句,它根据条件表达式 $(x > 0)$ 是否成立来决定是否执行其后的输入语句,若条件成立(即取值非 0)则从键盘输入缓冲区中读取一