

高等财经专科学校试用教材

# C 语言 程 序 设 计

褚东升 主编

东北财经大学出版社

图书在版编目(C I P)数据

C 语言程序设计/ 褚东升主编 .-大连:东北财经大学出版社,1996 .12

高等财经专科学校试用教材

ISBN 7-81044-169-8

.C... II 褚... III .C 语言-程序设计-高等学校:专业学校-教材 .TP312C

中国版本图书馆 CIP 数据核字(96)第 18799 号

高等财经专科学校试用教材

---

C 语言程序设计

---

C Yuyan Chengxu Sheji

---

1996 年 12 月第 1 版

褚东升 主编

1996 年 12 月第 1 次印刷

责任编辑:郭 洁

责任校对:孙 萍

---

出 版 者:东北财经大学出版社

地 址:大连·黑石礁

邮政编码:116025

---

制 版:大连斯达电脑开发公司

印 刷:

---

开 本:850 × 1168 1/32

印 张:8 1/2

字 数:213 000

印 数:1—5 000

---

ISBN 7-81044-169-8 T · 30

定价:12.00 元

# 前 言

为适应高等财经专科学校会计电算化专业教学的需要,根据财政部颁布的《财政(经)普通专科会计电算化专业教学方案》,财政部培训中心组织编写了会计电算化专业系列教材。本系列教材包括:《计算机基础》、《微型机原理与汇编语言》、《应用软件与信息安全管理》、《C 语言程序设计》、《FOXBASE+ 关系数据库系统》、《管理信息系统》、《会计电算化应用》等七本。该系列教材不仅适合于三年制大专会计电算化专业教学使用,也适合于成人教育及从事会计电算化工作人员学习参考。

《C 语言程序设计》由山东财政学院褚东升主编,并负责对全书的修订、总纂和定稿。全书写作分工如下:第一、七章由山东财政学院褚东升编写,第二、三章由江苏财政专科学校吴丹国编写,第四、六章由山东财政学院曲吉林编写,第五、八章由河南财税高等专科学校董汉丽编写。

在编写本书过程中,我们得到了财政部培训中心和兄弟院校的大力支持,山东财政学院的聂培尧教授在本书最初的体系安排及大纲编写等方面做了大量的工作,曲吉林副教授在本书的校稿中也做了大量的工作,在此表示衷心感谢!

本书及配套的教学大纲,已由财政部教材编审委员会组织的专家评审通过。

本书由中央财经大学潘省初教授主审。潘省初教授对全书提出了不少有益建议,在此深表谢意!

编 者

1996 年 6 月

## 内容提要

本书详细介绍了 C 语言的基本概念和用 C 语言进行结构化程序设计的方法,并配有大量的程序实例,说明了各种语法现象的特点及使用中应注意的事项。每章后均附有习题。

本书的特点是从介绍基础知识入手,采用深入浅出、循序渐进、逐步求精的程序开发技术介绍程序开发过程。本书体系合理,概念清晰,注重程序设计能力的训练。

本书可作为大中专院校程序设计课程的教材和教学参考书,也可作为计算机培训班的教材或参考书及计算机有关专业的程序设计课程的自学用书。

## 编 审 说 明

本书是全国财经类通用教材。经审阅,我们同意作为高等财经专科学校试用教材出版。书中不足之处,请读者批评指正。

财政部教材编审委员会

1996年8月29日

---

# 目 录

---

第一章 C语言概述 .....	1
§ 1.1 C语言的发展与特点 .....	1
§ 1.2 C语言的基本符号 .....	4
§ 1.3 简单的数据输入输出 .....	6
§ 1.4 C语言的程序结构 .....	14
§ 1.5 C语言程序的上机步骤 .....	16
第二章 基本数据类型、运算符和表达式 .....	21
§ 2.1 C语言的数据类型 .....	21
§ 2.2 常量和变量 .....	22
§ 2.3 整型数据 .....	25
§ 2.4 实型数据 .....	27
§ 2.5 字符型数据 .....	29
§ 2.6 变量的初始化 .....	32
§ 2.7 运算符与表达式 .....	33
第三章 程序控制语句 .....	49
§ 3.1 结构化程序设计概念及 C 语句 .....	49
§ 3.2 顺序结构 .....	53
§ 3.3 分支结构 .....	56
§ 3.4 循环结构 .....	66

---

---

§ 3.5	辅助语句 .....	73
§ 3.6	程序举例 .....	76

---

第四章	数 组 .....	84
-----	-----------	----

---

§ 4.1	数组的基本概念 .....	84
§ 4.2	一维数组 .....	85
§ 4.3	字符数组 .....	89
§ 4.4	多维数组 .....	92
§ 4.5	程序举例 .....	97

---

第五章	程序的构造及函数 .....	102
-----	----------------	-----

---

§ 5.1	C 语言的组成 .....	102
§ 5.2	C 语言的函数 .....	106
§ 5.3	函数的参数 .....	111
§ 5.4	函数的调用 .....	122
§ 5.5	程序编译预处理 .....	134

---

第六章	结构与联合 .....	142
-----	-------------	-----

---

§ 6.1	结 构 .....	142
§ 6.2	联 合 .....	156
§ 6.3	位字段 .....	161
§ 6.4	枚举类型 .....	163
§ 6.5	用 Typedef 定义类型 .....	167

---

第七章	指 针 .....	171
-----	-----------	-----

---

§ 7.1	指针的概念 .....	171
§ 7.2	变量的指针与指针变量 .....	172
§ 7.3	指针与数组 .....	181

---

§ 7.4 指针与函数 .....	196
§ 7.5 指针与结构 .....	207

---

第八章 文件及库函数的使用.....	222
--------------------	-----

---

§ 8.1 C 语言文件的概述 .....	222
§ 8.2 标准输入输出函数 .....	223
§ 8.3 标准设备文件及 I/O 改向 .....	227
§ 8.4 缓冲型文件输入输出系统 .....	230
§ 8.5 非缓冲型文件输入输出系统 .....	253
参考资料.....	261

# 第一章 C 语言概述

语言是一种通用性程序设计语言,它既具有高级语言特点,又具有低级语言特点的特性。随着计算机的广泛应用,C语言在各个领域的应用也越来越广泛。现在,几乎在各种型号的微型机和大、中、小型计算机上都配有C语言编译系统。C语言已广泛用于描述系统软件、数据处理、科学工程数值计算等多个领域,深受专业工作者和广大用户的欢迎。

## § 1.1 C语言的发展与特点

### 一、C语言的发展

早期的操作系统等系统软件主要是用汇编语言编写的,它依赖于计算机硬件,程序的可读性和可移植性都很差。为此,人们努力寻求一种程序语言,使它既具有高级语言的特性,能够编写可读性高、便于移植的程序,改善编写程序的环境;又具有某些必要的汇编语言特性,能描述对硬件的操作。例如能对内存地址的操作、对位的操作、对字的移位操作及对寄存器的操作等。C语言就是在人们寻找集高级语言和汇编语言优点于一身的高级语言过程中产生的。

C语言的前身是英国剑桥大学的 Martin Richards 在 60 年代开发的 BCPL 语言 (Basic Combined Programming Language)。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,开发了简单的、接近硬件的 B 语言(取 BCPL 的第一个字符),

并用 B 语言编写了 UNIX 操作系统和大量的实用程序。但是由于 B 语言是一种无类型面向机器字的语言,所以在描述各种数据结构时是很困难的。另外,由于 B 语言最后产生的是解释执行代码,运行速度较慢等原因而未能流行。此后,美国贝尔实验室的 Dennis M .Ritchie 对 B 语言作了进一步的充实和完善,于 1972 年推出了一种新型的程序设计语言——C 语言。随着 UNIX 操作系统在国际上的广泛流行,C 语言已被人们所接受,并在各种型号的微型机和大、中、小型计算机上都配有 C 语言编译系统。目前,C 语言已风靡全球,成为世界上应用最广泛的计算机程序设计语言之一。

1978 年,Brian Kernighan 和 Dennis M .Ritchie 合作编写了经典著作《The C Programming Language》,它是目前所有 C 语言版本的基础。1983 年,美国国家标准化协会(ANSI)对 C 语言的各种版本作了扩充和完善,制定了 C 语言的标准,称为 ANSI C。本书以 Turbo C 为线索介绍了 C 语言的特点和编程技术,Turbo C 完全是按照 ANSI 的 C 语言标准实施的,它是一种快速高效的编译程序。Turbo C 不仅提供了一个集成开发环境,同时还按传统方式提供了一个命令行编译程序版本,以满足不同用户的需要。

在 C 语言的基础上,1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言,成为一种面向对象的程序设计语言。C++ 提出了一些更为深入的概念,它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性,掌握起来有一定的难度。但是,C 语言是 C++ 的基础,C++ 语言和 C 语言在很多方面是兼容的。因此,掌握了 C 语言,再进一步学习 C++ 就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。

本书以目前较为流行的 Turbo C 2.0 为开发环境,介绍 C 语言及其程序设计。

## 二、C 语言的特点

C 语言具有多方面的特点,其主要特点有以下几个方面:

(一) C 语言不但具有高级语言的特点,而且还具有低级语言的特征,它允许直接访问地址,能进行位(Bit)运算,能实现汇编语言大部分功能,可以直接对硬件进行操作。

(二) 语言简洁、紧凑,使用方便、灵活。用 C 语言编写的程序通常比用其它高级语言编写的程序更简练,代码行少。语言的许多成分都通过函数调用完成,使得编译程序小而精。C 语言没有提供 I/O 操作、并行操作、同步或协同程序等复杂控制,而是提供了大量而有效的库函数来实现输入/输出、字符串处理及存贮分配操作等功能。C 语言运行时所要求的支持少,所以占内存空间也少。

(三) 丰富的运算能力。C 语言的运算符包含的范围很广泛,共有 34 种运算符。除一般高级语言使用的 +、-、\*、/ 四则运算及与(AND)、或(OR)、非(NOT)等逻辑运算功能外,还可以实现以二进制位(bit)为单位的位与(&)、位或(|)、位非(~)、位异或(^)以及移位(>>、<<)等位运算,并且具有如 a++、b--等单项运算和 +=、-=、\*=、/= 等复合运算功能。

(四) 丰富的数据类型。C 语言能在字符、整数、浮点数等基本类型的基础上,按层次逐层构造各种构造类型,如数组、指针、结构、联合等,能用来实现各种复杂的数据结构。因此,C 语言具有较强的数据处理能力。

(五) 结构化构造语言。C 语言是一种结构化程序设计语言,即程序的逻辑结构,可以用顺序、分支和循环三种基本结构组成。C 语言具有结构化控制语句,如 if...else、while、do...while、switch、for 等,十分便于采用由顶向下、逐步求精的结构化程序设

计方法。因此,用 C 语言编制的程序具有容易理解、便于维护等优点。

(六) 模块化的构造语言。C 语言程序的函数结构十分有利于把整体程序分割成若干个相对独立的功能模块,并且在程序间的相互调用、数据传递和数据共享方面提供了十分有效的手段。这一特点为把大型软件模块化、对由多人同时进行集体性开发的软件工程提供了强有力的支持。

(七) 预处理能力。C 语言中提供了 `# include` 和 `# define` 等编译预处理语句,能进行字符串或特定参数的宏定义,以及实现对外部文本文件的读取和合并。同时还具有 `# if`、`# else` 等条件编译预处理语句。这些功能的使用提高了软件开发的工作效率,并为程序的组织和编译提供了便利。

(八) 可移植性好。程序的可移植性是指在一个环境上运行的程序可以不加或稍加改动后在另一个完全不同的环境上运行。汇编语言是依赖于机器硬件的,用汇编语言编写的程序不可移植。而一些高级语言,它们从一种机器搬到另一种机器上,基本上都要根据国际标准进行重新编写。而 C 语言程序本身并不依存于机器硬件系统,从而便于在硬件结构不同的机种间和各种操作系统实现程序的移植。

## § 1.2 C 语言的基本符号

### 一、基本符号集

C 语言的基本符号是 ASCII 字符集。它包括:

(一) 数字:阿拉伯数字 10 个(0~9)。

(二) 字母:大写英文字母 26 个(A~Z)。

小写英文字母 26 个(a~z)。内部处理以小写字母为主。

(三)特殊符号:主要是指运算符和操作符,它通常是由 1~2 个特殊符号组成。共有以下 39 个:

+	-	*	/	%	<	< =	>	> =
= =	! =	& &		!	&		~	=
++	--	?:	<<	>>	<>	[]	.	->
+ =	- =	* =	/ =	% =	& =	^ =	=	,
^	#	sizeof						

## 二、标识符

C 语言的标识符可分为以下三类:

### (一) 保留关键字

C 语言每个关键字在 C 程序中都代表着某一固定含义,所有保留关键字都要用小写英文字母表示,且这些关键字都不允许作为用户标识符使用。C 语言共有以下 32 个保留关键字:

auto break case char const continue default do double else  
 enum extern float for goto if int long register return  
 short signed sizeof static struct switch typedef union un-  
 signed

viod volatile while

### (二) 预定义标识符

C 语言提供了大量的库函数和头文件,这些库函数名和头文件中定义的一些标识符都统称为预定义标识符。C 语言允许用户把这类标识符作其它用途,但这将使这些预定义的标识符失去原有的作用。例如:一旦用户把 Printf 说明为整型变量,则程序中不能再调用 Printf 进行输出了。鉴于目前各种计算机系统的 C 语言已经把这类标识符作为统一的库函数名或预编译处理中的专用命令名使用,因此为了避免误解,不要把这些预定义标识符另作它用或将它们重新定义。

### (三) 用户标识符

用户可以根据需要对 C 程序中用到的变量、符号常量、自己的函数或文件指针进行命名,形成用户标识符。这类标识符的构成规则如下:

1 .由英文字母、数字、下划线组成,且第一个字符不能是数字,必须是字母或下划线。

例如: a, b1, year \_ day, lines

2 .大小写字母的含义不同。

例如: VELOCITY, Velocity 和 velocity 是完全不同的标识符。

3 .一个名字可以由多个字符组成,但一般只有前八个字符有意义,可以相互区别。

例如: interest \_ Paid 和 interest \_ Payable 具有相同的意义。

### § 1 3 简单的数据输入输出

C 语言本身没有提供输入、输出操作的语句。C 语言程序中的输入和输出功能是通过调用系统提供的标准库函数实现的。本节先介绍两种经常使用的简单输入输出标准库函数,以利于程序举例,其余输入输出函数在第八章中介绍。

一、putchar( )函数和 getchar( )函数

putchar( )和 getchar( )是标准库函数。标准 I O 函数库中有一些公用的信息写在头文件 stdio .h 中,因此要使用标准 I O 函数库中的 I O 函数,一般应在程序开头写出下面的命令:

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

stdio .h 是 standard input & output 的缩写,称为标准输入/输出预说明, .h 是头文件的扩展名。它包含了与标准 I O 库有关的变量定义和宏定义。

(一) putchar( )函数

字符输出函数 `putchar( )` 的功能是把一个字符输出到标准输出设备上(通常是指显示器或打印机)。

其形式为: `putchar(ch);`

其中, `putchar( )` 是函数名, `ch` 是函数的参数, 在 `putchar( )` 中该参数必须是整型或字符型。该函数将 `ch` 的值以 ASCII 字符形式在标准输出设备上输出。

[例 1—1] `putchar( )` 函数的功能。程序如下:

```
#include <stdio.h>
main( )
{
    char a1, a2, a3;
    a1 = Y ; a2 = E ; a3 = S ;
    putchar(a1); putchar(a2); putchar(a3);
}
```

程序运行结果: YES

三个字符输出语句, 每个语句输出一个字符。

## (二) `getchar( )` 函数

字符输入函数 `getchar( )` 的功能是从标准输入设备上(通常是指键盘)读取一个字符, 该函数没有参数, 其函数值就是从输入设备得到的字符。

其形式为: `getchar( )`; 其中 `getchar` 是函数名。

[例 1—2] 输入一个字符。

```
#include <stdio.h>
main( )
{
    char c;
    c = getchar( );
    putchar(c);
}
```

```
}
```

程序运行时,如果从键盘上输入字符 Z 和回车:

```
Z
```

程序输出变量 C 的值 Z

```
Z
```

getchar( )函数只能接收一个字符,得到的是字符的 ASCII 代码,可以赋给一个字符型变量或整型变量,也可以不赋给任何变量只作为表达式的一部分。例如:

```
putchar(getchar( ))
```

就是以输入字符为参数用字符输出函数。

## 二、printf( )函数和 scanf( )函数

printf( )和 scanf( )函数是按指定的格式完成输入输出过程。前面介绍的字符输入、输出函数每次只能输入或输出一个字符,而 printf( )和 scanf( )函数一次可以输出或输入若干个任意类型的数据。虽然它们也是库函数,但在使用时不需要包含"stdio.h"。在 C 语言中,也就是这两个库函数在使用时不需要包含头文件库函数。

### (一)printf( )函数

格式输出函数 printf( )的功能是将一些变量的值按指定的格式输出到标准设备上。

其形式为:printf("格式控制",输出项表)。

其中:格式控制部分是一个用双引号括起来的字符串,用来确定输出项目的格式和需要原样输出的字符串;输出项可以是合法的常量、变量和表达式,输出项表中的各输出项之间要用逗号分开。

格式控制也称"格式控制字符串",它包括两类内容:格式说明和普通字符。

1 格式说明由"%"和格式字符组成,如:%d、%f等。它规定了输出项的输出格式。格式字符通常用于说明输出数据的类型。常用的格式说明中的格式字符及其意义如表 1—1 所示。

表 1—1 printf( ) 的格式字符

格式字符	说 明
d	以带符号的十进制形式输出整数(正数不输出符号)。
o	以八进制无符号形式输出整数(不输出前导符 o)。
x	以十六进制无符号形式输出整数(不输出前导符 ox)。
u	以无符号十进制形式输出整数。
c	以字符形式输出,只输出一个字符。
s	输出字符串。
f	以小数形式输出单、双精度数,隐含输出 6 位小数。
e	以标准指数形式输出单、双精度数,数字部分小数位数为 6 位。
g	选用 %f 或 e 格式中输出宽度较短的一种格式,不输出无意义的 0。

格式说明中,在 % 和上述格式字符间可以插入以下几种附加符号。如表 1—2 所示:

表 1—2 printf( ) 的附加格式说明字符

字符	说 明
l	表示输出的是长整型整数,可加在 d、o、x、u 前面。
m	表示输出数据的最小宽度。
n	对实数,表示输出 n 位小数,对字符串,表示截取 n 个字符。
0	表示左边补 0。
+	转换后的整数左对齐。
-	转换后的整数右对齐。

在使用 printf 函数输出时应注意:

- (1)数据类型应与上述格式说明匹配,否则将会出现错误。
- (2)格式字符必须用小写字母,如: %d 不能写成 %D。
- (3)可以在 printf( ) 函数中的"格式控制"字符串中包含第二