C 语言简明教程

陈　赞　　编著

张　晋　　审校

中国大地出版社

2001 年 • 北京

内容提要

随着 C 语言的不断普及和广泛应用，学习它的人越来越多。本书共组织了 9 个章节的内容，介绍了 C 语言的基本概念、语法规则和利用 C 语言进行程序设计的方法，并提供了大量实例和习题，是广大用户学习和使用 C 语言的良师益友。

本书可作为大专院校师生学习 C 语言的教材，也可供参加计算机等级考试的读者作为辅导教材。

丛书名：流行编程软件简明教程丛书

书名：C语言简明教程

作者：陈赞

出版社：海洋出版社

ISBN：7-5027-5325-7 / TP312C

出版日期：2001年1月

丛 书 前 言

计算机软件技术日新月异，编程软件种类繁多且各具特色。众多的计算机爱好者都希望能成为电脑编程的行家里手，如何才能向“卡乃基殿堂”迈出正确的第一步？

为满足各级 Windows 程序开发人员、大专院校相关专业的师生和业余编程爱好者学习和使用各种流行编程软件的需求，编者在搜集了不同层次读者意见的基础上，经过仔细探讨，精心策划了本套丛书，目的在于引导读者学习二十一世纪真正的编程工具！

本丛书名为“流行编程软件简明教程丛书”，主要包括：

《C++ Builder 简明教程》

《Delphi 简明教程》

《Visual C++ 简明教程》

《Visual Basic 简明教程》

《Visual FoxPro 简明教程》

《Quick BASIC 简明教程》

《C 语言简明教程》

本套丛书针对当前最流行的开发工具，通过浅显易懂、活泼生动的语言，采用逐步引导的方式将读者带入电脑编程的殿堂，使读者轻松掌握编程的基础知识，为进一步深造打下坚实基础。生动友好的例子帮助读者了解、掌握编程技巧，同时还大大激发了读者的兴趣。

本套丛书既不是面面俱到的“功能大全”，也不是单一查询函数的“用户手册”，而是独具特色的操作和编程指导书。丛书注重对软件的主要功能和新增特性进行介绍，结合实例和项目，讲解软件的主要功能、主要原理。

我们希望，本套丛书对于读者能“抛砖引玉、触类旁通”，指引读者踏上通往编程高手之路。

总的来说，本套丛书有以下几个最优越的特点：

- ✧ “精”——精选软件，精选作者，精选图书，是值得用户收藏的精品
- ✧ “准”——紧跟软件版本更新，出版时间准，图书出版市场定位准
- ✧ “简”——教程风格简单朴素，要让读者学习起来感觉简单实用
- ✧ “明”——丛书内容十分明晰，让读者读完之后能够真正明白

我们相信：

如果你是一位初学编程的读者，本套丛书将带领你入门！

如果你是一位对编程有些了解的业余爱好者，本套丛书为你指明前进的方向！

如果你是一位软件编程专业人员或计算机行家里手，本套丛书将为你的程序锦上添花！

编　者

2001 年 3 月

前　言

C 语言是目前国内外使用最广泛的程序设计语言之一。它处理功能丰富、表达能力强、使用方便灵活、执行程序效率高、可移植性强；既有高级语言的特点，又有汇编语言的特点。它具有较强的系统处理能力，可直接实现对系统硬件和外部接口的控制。它采用了自顶向下、逐步求精的结构化程序设计技术。另外，它的函数式结构也为实现程序的模块化提供了强有力的保障。因此，它被广泛地应用于系统软件和应用软件的开发。

由于 C 语言涉及的概念和规则比较多，使用起来虽然灵活但是容易出错，不少初学者感到困难，希望能有一本合适的教材，本书就是为了满足这一需求而编写的。

全书共分九章，并有 C 语言语法提要及系统函数库等附录。

第一章为 C 语言简介，叙述 C 语言的发展历程、主要特点、C 语言的结构特点以及 C 语言的编译和执行过程等内容。第二章介绍的是用 C 语言编程必须掌握的一些基础知识，包括最基本的数据类型、对数进行运算的运算符和数据的输入输出。第三章介绍了 C 语言程序的三种基本结构：顺序结构、选择结构、循环结构。

第四~七章分别讲述了数组、指针、函数和结构体、联合和枚举等内容，这部分是本书的重点章节，反映了 C 语言主要的特征。部分内容相对来说较难掌握，例如指向函数的指针、结构类型的递归应用、联合的概念及应用等。

第八、九章分别介绍了 C 语言的标准函数库和文件系统以及 C 语言的预编译程序两部分内容。

最后要感谢所有为编写本书提供帮助的人，他们是协助调试程序的周华、彭小琦以及为本书绘制了大量插图的高英丽同志。

由于作者水平有限，书中难免有错误和疏漏之处，恳请广大读者批评指正。

编　者

目　录

第一章　C 语言简介

C 语言是目前国际上广泛流行的一种结构化程序设计语言，它既适合作为系统描述语言，也可用来开发应用软件。因此，它深受广大程序设计者的欢迎。

本章，我们将带领读者学习 C 语言的发展史，并且掌握 C 语言程序的结构特点、C 语言程序的编译和执行等内容，为后面的学习打下基础。

1.1　C 语言的历史及特点

1.1.1　C 语言的历史

C 语言诞生之前，操作系统等系统软件主要是用汇编语言编写的（包括 UNIX 在内）。由于汇编语言依赖于计算机硬件，程序的可读性和可移植性都比较差。为了提高可读性和可移植性，最好改用高级语言，但一般高级语言难以实现汇编语言的某些功能（汇编语言可以直接对硬件进行操作，例如，对内存地址的操作、位操作等）。人们设想能否找到一种既具有一般高级语言特性，又具有低级语言特性的语言。于是，C 语言就在这种情况下应运而生了。

C 语言是在 20 世纪 70 年代初由美国贝尔实验室的 D.M.Ritchie 设计的，最初的 C 语言是为描述和实现 UNIX 操作系统而提供的一种工作语言。

到了 1973 年，K.Thompson 和 D.M.Ritchie 两个人合作把 UNIX 的 90%以上内容用 C 语言进行了改写，即 UNIX 第五版。C 是为开发 UNIX 操作系统而研制的，它随着 UNIX 的出名而闻名。C 语言的广泛应用又不断推出新的 C 语言版本，其性能也越来越强。

到了 1975 年 UNIX 第六版的推出和随着面向对象程序设计技术的出现，C 语言的突出优点引出了人们的普遍关注。

1978 年以后，C 语言逐渐风靡全世界，成为世界上应用最广泛的计算机语言之一。到目前为止，又发展到演变出了目前可在微机上运行的 Microsoft C/C++、Turbo C、Quick C、Borland C、Visual C/C++等版本。

1.1.2　C 语言的特点

C 语言之所以能被推广并被广泛使用，概括地说主要有如下特点：

- 语言简洁，使用方便。C 语言一共只有 32 个关键字，9 种控制语句，程序书写形式自由，主要用小写字母表示，压缩了一切不必要的成分。
- 运算符丰富。C 语言的运算符包含的范围很广泛，共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 语言的运算类型极其丰富，表达式类型多样化，灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

- 数据结构丰富，具有现代化语言的各种数据结构。C 语言的数据类型有：整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据结构（如链表、树、栈等）的运算。尤其是指针类型数据，使用起来比其他高级语言更为灵活、多样。
- 具有结构化的控制语句（如 if…else 语句、while 语句、do…while 语句、switch 语句、for 语句）。用函数作为程序模块以实现程序的模块化。是结构化的理想语言，符合目前通行的编程风格要求。
- 语法限制不太严格，程序设计自由度大。例如，对数组下标越界不作检查，由程序编写者自己保证程序的正确。对变量的类型使用比较灵活，例如，整型变量与字符型数据以及逻辑型数据可以通用。一般的高级语言语法检查比较严，能检查出几乎所有的语法错误，而 C 语言允许程序编写者有较大的自由度，程序员应当仔细检查程序，保证其正确，而不要过分依赖 C 语言编译程序去查错。“限制”与“灵活”是一对矛盾。限制严格，就失去灵活性；而强调灵活，就必然放松限制。一个不熟练的人员，编一个正确的 C 语言程序可能会比编一个其他高级语言程序难一些。也就是说，对用 C 语言的人，要求对程序设计更熟练一些。
- C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此 C 语言既具有高级语言的功能，又具有低级语言的许多功能，可用来写系统软件。C 语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言。有人把 C 语言称为“高级语言中的低级语言”，也有人称它为“中级语言”，意为兼有高级和低级语言的特点。
- 生成目标代码质量高，程序执行效率高。一般只比汇编程序生成的目标代码效率低 10%~20%。
- 用 C 语言写的程序可移植性好（与汇编语言比）。基本上不作修改就能用于各种型号的计算机和各种操作系统。

C 语言的以上特点，读者现在也许还不能深刻理解，待学完 C 语言以后再回顾一下，就会有比较深的体会。

我们从应用的角度出发对 C 语言和其他高级语言作一简单比较：

从掌握语言的难易程度来看，C 语言比其他语言难一些，BASIC 是初学者入门的较好的语言，FORTRAN 也比较好掌握。

科学计算多用 FORTRAN；对商业和管理等数据处理领域，用 COBOL 为宜，C 语言虽然也可用于科学计算和管理领域，但并不理想，C 语言的特长不在这里。

对操作系统和系统实用程序以及需要对硬件进行操作的场合，用 C 语言明显地优越于其他高级语言，有的大型应用软件也用 C 语言编写。

从教学角度，由于 PASCAL 是世界上第一个结构化语言而曾被认为是计算机专业的比较理想的教学语言，目前在“数据结构”等课程中一般用 PASCAL 语言举例，但 PASCAL 语言难以推广到各实际应用领域，到目前为止基本上只是教学语言，C 语言也是理想的结构化语言，用描述能力强，同样适于教学，而且“操作系统”课程多结合 UNIX 讲解，而 UNIX 与 C 语言不可分，因此，C 语言有可能取代 PASCAL 而成为被广泛使用的教学语言，而且 C 除了能用于教学外，还有广泛的应用领域，因此更有生命力。

PASCAL 和其他高级语言的设计目标是通过严格的语法定义和检查来保证程序的正确性，而 C 则是强调灵活性，使程序设计人员能有较大的自由度，以适应宽广的应用面。

总之，C 语言对程序员要求较高。程序员使用 C 语言编写程序会感到限制少、灵活性大、功能强，可以编写出任何类型的程序。现在，C 语言已不仅用来编写系统软件，也用来编写应用软件。

1.2　C 语言程序的结构特点

任何一种计算机语言，都有特定的语法规则和表现形式，C 语言也不例外。程序的构成规则和书写格式则是其表现形式的重要方面，下面就来介绍 C 语言最基本的结构特点。

1.2.1　构成 C 语言的基本字符和标识符

C 语言规定了其所需的基本符号和标识符，这些是初学者首先应掌握的。

1. 字符集

满足 C 语言语法要求的字符集如下：

- 英文字母 a～z，A～Z；
- 阿拉伯数字 0～9；
- 特殊符号（如表 1-1 所示）。

表 1-1　C 语言中可以使用的特殊符号

+	-	*	/	%	_下划线	=	<
>	&	~	（	）	[	]	.
{	}	:	?	;	"	!	#
空格	'	|	^				

2. 标识符

C 语言的标识符主要用来表示常量、变量、函数和类型等的名字，是只起标识作用的一类符号。它包括如下 3 类：

（1）保留字

所谓保留字，就是这样一类标识符，其每一个都有特定的含义，不允许用户把它们当作变量名使用，C 语言的保留字都用英文小写字母表示，共有 33 个保留字，如表 1-2 所示。

表 1-2　C 语言的保留字

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

（2）预定义标识符

除了上述保留字外，还有一类具有特殊含义的标识符，它们被用作库函数名和预编译命令，这类标识符在 C 语言中称为预定义标识符。一般来说不要把标识符再定义为其他标识符（用户定义标识符）使用。预定义标识符包括预编译程序命令和 C 编译系统提供的库函数名。

其中预编译程序命令有：define、undef、include、ifdef、ifndef、endif、line。

（3）用户定义标识符

用户定义标识符是程序员根据自己的需要定义的一类标识符，用于标识变量、符号常量、用户定义函数、类型名和文件指针等。这类标识符主要由英文字母、数字和下划线构成，但开头字符一定是字母或下划线。下划线（_）起到字母的作用，它还可用于一个长名字的描述，如：

```
numbergoodstudent
```

可写为：

```
number_good_student
```

上面的写法中，用下划线把名词隔开，以增加可读性。

在 C 语言中，大小写字母的变量含义是不同的，如 TOTAL、Total、…、total 等是完全不同的名字。通常变量名用小写字母，常数名用大写字母。一个变量名字可由许多字符组成，但其长度是有限的，对于 ANSIC 只有前 31 个字符有效。对旧标准是前 8 个字符有效，例如 student_AAA 和 Student_BBB 编译程序把它们视为同一个名字。

为了使程序清晰、易读，建议在定义标识符时，应注意如下几点：

- 名字要有明确的含义，应尽量选用具有一定含义的英文单词来命名，使读者“见其名而知其意”。例如，代表总和的标识符用 sum 要比用 st 好，代表平均数的标识符用 average 而不用 a 等。如果所选用的英文单词太长，可采用公认的缩写方式。例如，圆周率用 PI 来命名。
- 标识符一般采用常用取简、专用取繁的原则。即常用的标识符应当定义为既简单又易识别的符号。
- 对于由多个单词描述的标识符，建议用下划线将各单词隔开，以增强可读性。例如，average_salary。
- 对于标识变量的标识符，可用特定的字符作其前缀来表示变量的数据类型。例如，用“i”表示整数、“I”表示长整数、“c”表示字符型、“sz”表示串类型等。

1.2.2　C 语言程序的实例

为了说明 C 语言程序的结构特点，先看几个简单的 C 语言程序实例，以便使读者有一个初步的认识。

【例 1-1】编写显示字符串“Hello! C Program”的 C 语言程序。

具体程序代码如下：

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

这是一个最简单的 C 语言程序，它把字符串“Hello! C Program”显示在屏幕上。该程序由一个函数 main（）（叫主函数）构成。任何一个程序都必须有此函数，花括号{}所括的内容是 main 的函数体，每个 C 语言程序的函数都至少有一对{}。

printf（）是由系统提供的标准库函数，它完成输出功能，C 语言的输出是由函数来完成的，而与系统无关，这是它的特点之一。“Hello! C Program”是要输出的内容。“\ n”表示换行字符，它是由“\”和“n”二字符构成，属转义字符，有关转义字符，在后面将会具体介绍。printf（）后的分号是语句结束符，C 的每一个语句都以“;”终止。

#include 是预编译程序命令，它把头文件“stdio.h”的内容展开在#include <stdio.h>所在的行位置处。其中，#include <stdio.h>也可以写成#include“stdio.h”，“stdio.h”文件中定义了 I/O 库所用到的某些宏和变量。因此，在每一个引用标准库函数的程序中都必须带有该#include <stdio.h>命令行。

【例 1-2】计算两个数之和的 C 语言程序。

具体程序代码如下：

```
# include <stdio.h>                          / * 计算两数之和的 C 语言程序 * /
main（）
 {
     float a，b，c;                          / * 定义 a，b，c 的数据类型为实型 * /
     printf（"Please input the two datas \ n："）;
     scanf（" % f % f"，&a，&b）;              / * 输入 a，b 两个数 * /
     c = a + b;                              / * 求和 * /
     printf（" \ n sum = % f \ n"，c）;
 }
```

运行该程序时，首先提示输入两个数 a 和 b，然后计算出它们的和，并把结果以如下形式显示在屏幕上：

```
     sum=…
```

在此程序中，/ * … * /是一个注释语句，其中包含注释的内容，它在程序的编译过程中不产生任何执行代码，只是在编程中起到备忘录的作用。“float a，b，c;”是数据类型说明语句，它把 a、b 和 c 定义为实型数。值得注意的是，C 语言程序中的变量，在使用之前都要定义其数据类型。

“scanf（…）;”是输入语句，scanf（）是格式化输入函数，它是一个由系统提供的标准库函数，其后的括弧内为参数表，“% f % f”为格式串，% f 表示实型数格式，指明给 a、b 等要求输入实型数。执行该语句时，数据从键盘上输入。

“c=a+b;”是赋值语句（或表达式语句），等号（＝）是赋值运算符，表示把右边表达式的运算结果赋给 average。

“printf（“\ n sum=% f \ n”，c）;”为输出语句，它首先在新的一行输出字符串“sum＝”，然后按实型数格式（% f）输出变量 c 的值，并使光标移至下一行。

【例 1-3】求 a 和 b 两个数中的较大的值。

具体代码如下：

```
# include <stdio.h>                          / * 计算较大值的 C 语言程序 * /
main（）                                     / * 主函数 * /
 {
     int a，b，imax;                         / * 定义变量类型 * /
```

```
    printf（"please input two datas a，b：\n"）；
    scanf（"%d%d"，&a，&b）；          /* 在此输入 a，b 的值 */
    imax = max（a，b）；               /* 调用求最大值的函数 */
    printf（"\n maximum is %d"，imax）；   /* 输出结果 */
}
int max（x，y）；                      /* 求最大值函数 */
int x，y；
{
    int m；
    if（x>y）
        m = x；
    else
        m = y；
    return（m）；                      /* 向主程序返回最大值结果 */
}
```

此程序由两个函数组成，除了主函数 main（）之外，还有一个计算最大值的函数 max（）。“int max（x，y）”说明函数的返回值类型为 int（整型），函数名字为 max，函数的参数为 x，y。“int x,y;”说明各参数的类型，“return（m）”将求解结果返回给主函数。

该程序的执行是从 main（）函数开始，当主函数执行到 imax=max（a，b）语句时，控制被传递给 max（）函数，当执行 return（m）语句时，则结束 max（）函数，控制又被传递给 main（）函数，并把 max（）的计算结果带给 main（）函数。当主函数执行结束时，整个程序的执行也就结束了。

1.2.3　C 语言程序的结构特点

由上面几个简单的 C 语言程序实例，可以看出 C 语言程序的结构有如下几个特点：

1. C 语言程序由一个或多个函数组成

其中必须有一个主函数，主函数名为 main。其余函数的名字由程序设计者自定。

程序的执行是从主函数开始，其他函数都是在开始执行 main 函数以后，通过函数调用或嵌套调用而得以执行的。主函数是整个程序的控制部分。

主函数以外的其他函数可以是系统提供的库函数，也可以是用户根据自己的需要而编制的函数。为了便于程序设计，各种 C 语言的版本都提供了大量的库函数，供程序设计者引用。

2. 函数组成

C 语言函数的定义包括函数说明和函数体两个部分。函数说明指明函数的类型、属性、函数名、参数和参数说明等，如例 1-3 中的 max 函数的说明部分为：

```
int max（x，y）；
int x,y；
```

函数体是花括号所括的部分，它包括局部变量的说明语句一组执行语句。每个语句都由分号“；”结束。综上所述，一般函数的结构如下：

```
数据类型标识符 函数名（形参表）
形参说明：
{
    局部变量说明语句；
    执行语句；
}
```

3. 外部说明

在函数定义之外还可包含一个说明部分，该说明部分叫外部说明，它可包括预编译命令（如例 1-3 中的#include）、外部变量的说明等。

1.3　C 语言程序的编译和执行

当把 C 语言程序编写好之后，就可以在机器上运行它了。我们把编写好的 C 语言程序叫 C 源程序。由于 C 语言是一种高级程序设计语言，它很容易被人们看懂和接受，但是，对于计算机来说，却不能接受这种语言，它只能接受机器语言。为此，首先必须把 C 语言程序翻译成相应的机器语言程序，这个工作叫编译。

1. 源文件的编辑

为了编译 C 源程序，首先要用系统提供的编辑器建立一个 C 语言程序的源文件。一个 C 语言源文件是一个编译单位，它以文本格式存放在计算机的文件系统中（硬盘上）。源文件名自定，文件的扩展名（或后缀名）为“.c”。例如：

```
myfile.c
file.c
```

一个大的 C 语言程序往往可划分为若干模块，每个模块由不同的人或小组负责编写。对每个模块可建立一个源文件。因此，一个大的 C 语言程序可包含多个源文件。

2. 编译

源文件建立好后，经检查无误后就可进行编译。编译是由系统提供的编译器完成，编译命令随系统的不同而异，具体操作时可参考相应的系统手册。例如，对于 Turbo C，一般通过 Turbo C 的编辑环境界面中的 Compile 菜单中的 Compile 命令进行编译，编译器在编译时对源文件进行语法和语义检查，并给出所发现的错误。用户可根据错误情况，使用编辑器进行修改，然后对修改后的源文件再度编译。用户也可以在 Compile 菜单中选 Make 命令进行编译，它能直接生成可执行的文件，此时如果系统发现用户的程序有语法错误，就发出错误的参考信息，提示用户进行错误代码的修改，然后用户再重新进行编译。

3. 连接编辑

在上述步骤中，若用户选择 Compile 命令进行编译时，编译所生成的目标文件（*.obj）是相对模块，还不能直接执行，必须用连接编辑器把它和其他目标文件以及系统所提供的库函数进行连接装配，生成可执行文件存于文件系统中。可执行文件的名字可自由指定，扩展名为“.exe”。如图 1-1 所示，是 C 语言程序的操作过程。