高职高专 现代信息技术系列教材

C语言程序设计教程

宗大华　陈吉人　编

人民邮电出版社
POSTS & TELECOM PRESS

高职高专现代信息技术系列教材

C语言程序设计教程

宗大华　陈吉人　编

人民邮电出版社

图书在版编目（CIP）数据

C 语言程序设计教程/宗大华，陈吉人编．—北京：人民邮电出版社，2004.6
（高职高专现代信息技术系列教材）
ISBN 7-115-12244-X

I．C… II．①宗…②陈… III．C 语言—程序设计—高等学校：
技术学校—教材 IV．TP312
中国版本图书馆 CIP 数据核字（2004）第 031515 号

内 容 提 要

本书是为高职高专学生编写的 C 语言教材。全书共分为八章：概述、数据类型、运算符与表达式、三种基本的语句结构、数组、指针、函数、用户自定义的数据类型，以及 C 的文件操作函数。

本书力求使初学者能够建立正确的 C 语言概念，学会基本的编程方法，形成对 C 语言的一个整体了解。书中安排了大量的示例，每章的后面都配有适量的练习题。认真地阅读、理解和完成它们，肯定会对读者认识、掌握，进而领悟用 C 语言解决实际问题的方法带来裨益。

高职高专现代信息技术系列教材

C 语言程序设计教程

◆ 编　　宗大华　陈吉人
责任编辑　潘春燕
执行编辑　韩学义

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京铭成印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：17.25
字数：413 千字　　2004 年 6 月第 1 版
印数：10 001 – 13 000 册　　2006 年 2 月北京第 4 次印刷

ISBN 7-115-12244-X/TP・3955

定价：23.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223

高职高专现代信息技术系列教材

编 委 会 名 单

丛书前言

江泽民总书记早在十五大报告中提出了培养数以亿计高素质的劳动者和数以千万计专门人才的要求，指明了高等教育的发展方向。只有培养出大量高素质的劳动者，才能把我国的人数优势转化为人才优势，提高全民族的竞争力。因此，我国近年来十分重视高等职业教育，把高等职业教育作为高等教育的重要组成部分，并以法律形式加以约束与保证。高等职业教育由此进入了蓬勃发展时期，驶入了高速发展的快车道。

高等职业教育有其自身的特点。正如教育部“面向 21 世纪教育振兴行动计划”所指出的那样，“高等职业教育必须面向地区经济建设和社会发展，适应就业市场的实际需要，培养生产、管理、服务第一线需要的实用人才，真正办出特色。”因此，不能以本科压缩和变形的形式组织高等职业教育，必须按照高等职业教育的自身规律组织教学体系。为此，我们根据高等职业教育的特点及社会对教材的普遍需求，组织高等职业学校有丰富教学经验的老师，编写了这套《高职高专现代信息技术系列教材》。

本套教材充分考虑了高等职业教育的培养目标、教学现状和发展方向，在编写中突出了实用性。本套教材重点讲述目前在信息技术行业实践中不可缺少的、广泛使用的、从业人员必须掌握的实用技术。即便是必要的理论基础，也从实用的角度、结合具体实践加以讲述。大量具体的操作步骤、许多实践应用技巧、接近实际的实训材料保证了本套教材的实用性。

在本套教材编写大纲的制定过程中，广泛收集了高等职业学院的教学计划，调研了多个省市高等职业教育的实际，反复讨论和修改，使得编写大纲能最大限度地符合我国高等职业教育的要求，切合高等职业教育实际。

在选择作者时，我们特意挑选了在高等职业教育一线的优秀骨干教师。他们熟悉高等职业教育的教学实际，并有多年的教学经验；其中许多是“双师型”教师，既是教授、副教授，同时又是高级工程师、认证高级设计师；他们既有坚实的理论知识，很强的实践能力，又有较多的写作经验及较好的文字水平。

目前我国许多行业开始实行劳动准入制度和职业资格制度，为此，本套教材也兼顾了一些证书考试（如计算机等级考试），并提供了一些具有较强针对性的训练题目。

对于本套教材我们将提供教学支持（如提供电子教案等），同时注意收集本套教材的使用情况，不断修改和完善。

本套教材是高等职业学院、高等技术学院、高等专科学院教材。适用于信息技术的相关专业，如计算机应用、计算机网络、信息管理、电子商务、计算机科学技术、会计电算化等。也可供优秀职高学校选作教材。对于那些要提高自己应用技能或参加一些证书考试的读者，本套教材也不失为一套较好的参考书。

最后，恳请广大读者将本套教材的使用情况及各种意见、建议及时反馈给我们，以便我们在今后的工作中，不断改进和完善。

关于本书

这是一本为高职高专学生编写的C语言教材。

作为一种程序设计语言，C语言语句简洁、使用灵活、性能高效、应用面广，有着自己独特的魅力。它已成为计算机及相关专业人员学习计算机程序设计的首选语言。

针对读者群，本书在内容上做了取舍，去除了一些较难理解的或非基本的知识。全书共分为8章。

第1章是概述。希望通过它，能够在读者的脑海里对如何用C语言编写程序，形成一个初步的印象。第2章讲述数据类型、运算符与表达式。它是学习C语言的基础。因为计算机所做的事情，无外乎就是接收输入数据，进行数据处理，然后输出结果。第3章讲述用C语言编写程序时的3种基本语句结构。人们正是运用这些语句结构，才能够编写出各式各样的程序。前3章内容可以归为本书的基础篇。从第4章的数组开始，随后的第5章指针、第6章函数、第7章用户自定义的数据类型，以及第8章C的文件操作函数，应该是本书的提高篇。有了这前后的8章知识，读者就具备了用C语言编写程序的基本能力和手段了。

本书在编写时，着力于以下几点：

（1）力求以严谨而通俗的语言，讲述C语言的语法，以使初学者能够建立起正确的概念，掌握语言本身的特征。

（2）力求通过丰富的示例，讲述C语言的编程技术，以使初学者能够领略其中的真谛，学会基本的编程方法。

（3）力求用程序运行的结果，验证C语言语法的各种规定，以使初学者能够明了事物的本质，形成对C语言的一个整体了解。

希望读者在学习计算机程序设计语言的过程中，一定要以极大的热情去动手实践。在实践过程中一定要有百折不挠的精神，一定要深信“失败是成功之母”！只有坚持上机实践，不断体验失败，继而不断迎取胜利，才有可能达到成功的顶峰。

在本书的编写过程中，宗涛、蒋玮和梁发寅为每章所附习题的收集、调试做了很多的工作，在此表示诚挚的谢意。由于作者水平有限，书中难免出现不当甚至错误之处。恳请广大读者批评、指正。

编　者

2004年3月于北京

目　录

第 1 章　概述

“由表及里”是一种认识问题的方法。这一章就是先从外表来看一下 C 语言，以便能在人们的脑海里，对 C 语言留下一个初步的印象：什么是计算机程序？什么是计算机的程序设计语言？有哪几种程序设计语言？C 语言在其中处于什么位置？用 C 语言编写的程序大致是个什么样子？在什么环境里能够编写 C 语言的程序？有了这些初步的印象，你就会知道应该如何去学习 C 语言，知道在学习过程中把自己的注意力集中在什么地方。这一切，对于你学习、掌握 C 语言，肯定是至关重要的。

本章着重讲述 4 个方面的内容：

（1）C 语言程序的基本组成。

（2）C 语言的基本词法（字符集、保留字和标识符的构成）。

（3）用 C 语言编写程序时的四项工作。

（4）Turbo C 开发环境简介。

1.1　高级语言与 C 语言

1.1.1　程序设计语言与 C 语言

按照字典的说法，“程序”是事情进行的先后次序。因此，“计算机程序”即是要让计算机去完成的事情的先后次序。要让计算机去完成的事情，当然是由人去交给计算机做的。这就是说，人要使用一种办法，把自己要计算机去做的事情描述出来，然后才能提交给它去做。通常，把人与计算机之间交换信息的工具，称之为“计算机程序设计语言”。人们就是用计算机程序设计语言来编写计算机程序，然后交于计算机去执行的。

自世界上第一台计算机在 1946 年问世以来，用于编写计算机程序的程序设计语言，由机器语言发展到汇编语言，又由汇编语言发展到高级语言。

“机器语言”是指计算机本身自带的指令系统。计算机的指令由二进制数序列组成，用来控制计算机进行某种操作。指令由操作码和地址码两部分组成。其中，操作码规定计算机要做的运算；地址码告诉计算机是由哪些数来参加运算，在什么地方能找到那些数，计算完毕后的结果应该存放到哪里去等等。很明显，用机器语言编写的程序，不必通过任何翻译处理，计算机硬件就能够直接识别和接受。因此，用机器语言编写的程序，具有质量高、执行速度快和占用存储空间少等优点。但是，它缺乏直观性，难学、难记、难检查以及难修改。

为了克服机器语言的缺点，出现了汇编语言。汇编语言是一种面向机器的程序设计语言。也就是这种语言的指令基本上与机器指令一一对应。不过，在汇编语言中，用助忆符（一种便于记忆的符号）代替了机器指令中的操作码，用符号地址代替了机器指令中的地址码。正是这种代替，使得机器语言得以“符号化”。比起机器语言来，它好记了，读起来容易了，检查、修改也方便了。但是这样一来，用汇编语言编写的程序，计算机却不能直接识别和接受，它必须要由一个起翻译作用的程序将其翻译成机器语言程序，这样计算机才能执行。这个起翻译作用的程序，通常被称为“汇编程序”，这个翻译过程，称之为“汇编”。

汇编语言的缺点是依赖于具体的机器（所以前面说它是面向机器的），不具有通用性和可移植性。另外，与人们习惯使用的自然语言和数学语言相差甚远，因此又出现了所谓的高级语言。

高级语言是一种很接近于人们习惯使用的自然语言（即人们日常使用的语言）和数学语言的程序设计语言。在用高级语言编写计算机程序时，允许出现规定的英文词汇；在书写计算式子时，所用的运算符号和组成的算式，与我们日常用的数学式子差不多。因此，人们用它来编写计算机程序，比起使用机器语言和汇编语言，显然要方便得多。

用高级语言编写的程序，称为“源程序”。如同用汇编语言编写的程序，计算机不能直接识别一样，用高级语言编写的程序，计算机也不可能直接识别与接受。因此也必须要有一个“翻译”，把源程序翻译成机器指令的程序，然后再让计算机去执行这个机器语言程序。对于高级语言来说，翻译过程有两种方式：一种是事先编好一个称为“编译程序”的机器指令程序，它把用高级语言编写的源程序整个地翻译成用机器指令表示的机器语言程序（这个由编译程序翻译出来的结果程序，称为“目标程序”），然后执行该目标程序。这种翻译过程如图 1-1（a）所示。另一种是事先编好一个称为“解释程序”的机器指令程序，它把用高级语言编写的源程序逐句翻译，译出一句就立即执行一句。这种翻译过程如图 1-1（b）所示。前一种翻译方式称为“编译”方式，后一种翻译方式称为“解释”方式。

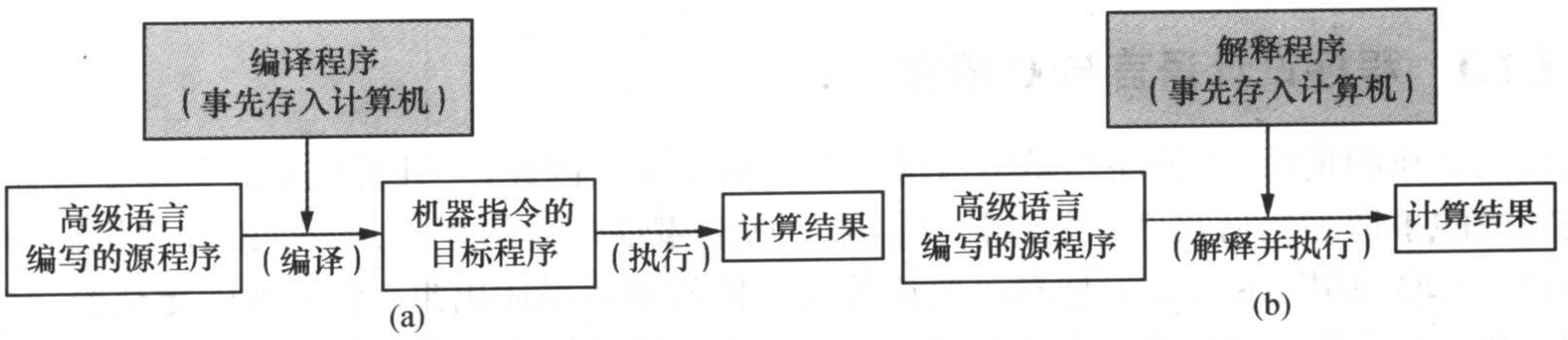

图 1-1　编译和解释两种翻译方式示意图

C 语言就是一种高级语言，它用比较接近人的思维和表达问题方法的形式来描述问题、编写计算机程序，然后以编译的方式进行翻译。

例 1-1　分别用机器语言、汇编语言和 C 语言描述算式：z=x+y。

解：描述两数相加，对于机器语言，可以有如下程序段：

```
A11001
03062001
A33001
```

它表示先把 0110 地址单元中所存的数取到寄存器 AX；然后把 0120 地址单元中的数取

出，与 AX 里的内容相加，结果在 AX 中；最后把 AX 里的结果存到 0130 地址单元。

对于汇编语言，可以有如下程序段：

```
MOV AX，[0110]
ADD AX，[0120]
MOV [0130]，AX
```

对于 C 语言，可以有如下程序段：

```
int x=235, y=368;
z=x+y;
```

它表示让变量 x 和 y 分别取值 235 和 368，求和后把结果存放在变量 z 里。编程者并不需要去管数据放在何处，就像是写一道算术题似地。

从解答中可以看出，机器语言程序完全没有直观性可言，如果你不了解机器指令 A1 是表示将跟随其后单元中的内容送寄存器 AX，那么你根本无法知道它的含义。对于汇编语言，MOV 就是英文 move 的缩写，因此可以知道它是要把一个数据送到寄存器 AX 里面去。可见，汇编语言具有一定的直观性，便于人们记忆。再看 C 语言，它简直就是近乎于直接使用人们习惯的数学表达式来描述加法。可见，学习用 C 语言来编写计算机的程序，人们容易接受。

1.1.2　简单的 C 语言程序

先让我们通过两个简单的 C 语言程序，大致领略一下用 C 语言来编写程序时通常的做法，并总结出 C 语言程序的一些特点。

例 1-2　用 C 语言编写一个程序，它接收从键盘输入的两个整数，求和后打印输出。

解：程序编写如下：

```
#include "stdio.h"
main()
{
   int m, n, sum;                    /* 变量说明 */
   scanf ("%d%d", &m, &n);           /* 从键盘输入数据 */
   sum=m+n;                          /* 求和 */
   printf ("sum=%d\n", sum);         /* 打印输出 */
}
```

在 C 语言中，以符号“/*”开始、“*/”结束的中间部分，是对左边程序语句的注释。不难通过上面程序中给出的注释，读懂花括号里整个程序的安排。

第 1 条语句：

```
int m, n, sum;
```

表示 m、n 和 sum 是 3 个变量，前面的 int 说明它们都是整型的（单词 integer 的缩写）。

第 2 条语句：

```
scanf ("%d%d", &m, &n);
```

是一条格式输入语句。其中&m 和&n，表示这两个变量所对应的内存单元地址。这条语句的功能是按照格式符“%d”的规定，从键盘接收两个十进制的输入数据，分别存放到地址&m 和&n 指定的存储单元中。

第 3 条语句：

sum=m+n;

是把 m 和 n 相加后，将求得的和存入变量 sum 中保存。注意，C 语言中的符号“=”，不是等号，而是赋值运算符，表示是把右端的计算结果送给左端的变量。

第 4 条语句：

printf ("sum=%d\n", sum);

是一条格式打印输出语句，即将变量 sum 的当前值，按照格式符“%d”的规定（十进制整数）的形式输出。比如说，如果现在从键盘上输入的两个数是 3 和 5，那么，在显示器上就会输出信息：sum=8。

例 1-3 用 C 语言编写一个程序，它接收从键盘输入的两个整数。比较后，将其中的大者打印输出。

解：在日常生活中，人们总是把大的、复杂的事情，化为若干小的、简单的事情去处理。在进行程序设计时，也常采用这种方法。在这个程序里，我们把接收键盘的输入和打印输出作为一件事情来处理（表现在 main()里），把判断两个数的大小作为另一件事情来处理（表现在 max()里）。然后通过一定的办法，把这两件事情拼接到一起，整个事情也就处理了。程序编写如下：

```
#include "stdio.h"
int max (int x, int y)
{
   int z;
   if (x>y)
      z=x;
   else
      z=y;
   return (z);
}
main ()
{
   int a, b, c;
   scanf (" %d%d ",&a, %b);
   c=max (a, b);
   printf (" max=%d\n",c);
}
```

对函数max的调用

从函数 max 返回

从程序中可以看出，它由两个函数组成：一个名为 main，一个名为 max。在函数 main 里，使用格式输入语句 scanf，往变量 a 和 b 里输入数据，然后用 a 和 b 去调用函数 max。max 的功能是比较两个数的大小，把大者存入变量 z，通过 return 语句，把 z 的值返回。这样，从函数 max 返回时，就会把 z 里比较结果的值送给函数 main 里的变量 c。最后，由格式打印语句 printf，把变量 c 的内容打印出来。

由以上两个例子，可以初步归纳出用 C 语言编写计算机程序时，具有如下特点。

（1）C 语言程序都是由一个个函数组成的，函数是 C 语言程序的基本单位。比如，例 1-2 的程序，是由一个名为 main 的函数组成的；例 1-3 的程序，是由一个名为 main 的函数和一个名为 max 的函数组成的。

（2）每一个 C 语言程序，都有一个、且只有一个名为 main 的主函数，整个程序从它开始执行。至于 main 函数在整个程序中所放的位置，与它作为程序开始执行的地位没有什么关系。也就是说，main 函数可以安排在整个程序的最前面、中间或后面。

（3）C 语言程序中的每一个语句，都以分号作为自己的结束。也就是说，在 C 语言中，分号“;”是一个语句的结束标志。

（4）在 C 语言程序中，可以用/*……*/形成注释，以对程序中的所需部分做出说明。/*是注释的开始符，*/是注释的结束符，必须配对使用。

1.1.3　程序设计时的算法描述

用计算机程序设计语言编写程序，首先应该选定要用的计算公式，制定解决问题的步骤，确定程序采用的结构（到第 3 章时会知道，程序的结构主要有 3 种形式：顺序结构、选择结构以及循环结构）等等，然后，才能真正动手去编写程序和上机。这个在真正动手之前的准备环节，就是所谓的算法描述阶段。这个阶段，对于问题的解决，无疑是非常重要的。

为了把解决问题的方法和步骤（也就是所谓的算法）描述出来，可以借助于人们日常使用的语言（称为“自然语言”）；可以借助于传统的流程图；可以借助于 N-S 流程图；也可以借助于介于自然语言和计算机语言间的文字和符号（称为“伪代码”）。总之，描述的方法是多样的，目的只有一个，即按照算法的描述编写程序时，思路会更加清晰。

在后面学习编写程序时，如果认为有必要，我们会用自然语言对程序的编写步骤加以讲述，解释使用的计算公式，以帮助对所编程序的理解。图 1-2 里给出传统流程图的一些常用符号，在第 3 章介绍程序结构时，会涉及到它们。

比如可以利用这些符号，为例 1-3 中的进行“判断两个数大小”的函数 max 绘制流程图，如图 1-3 所示。图中清楚地反映出“x>y?”是一个条件。如果条件成立（yes），那么就去做“z=x”；否则（no）就去做“z=y”。这是一种根据条件做出选择的流程图，它有两个出口：yes 和 no。编程时，就应该使用双分支选择语句去应对。

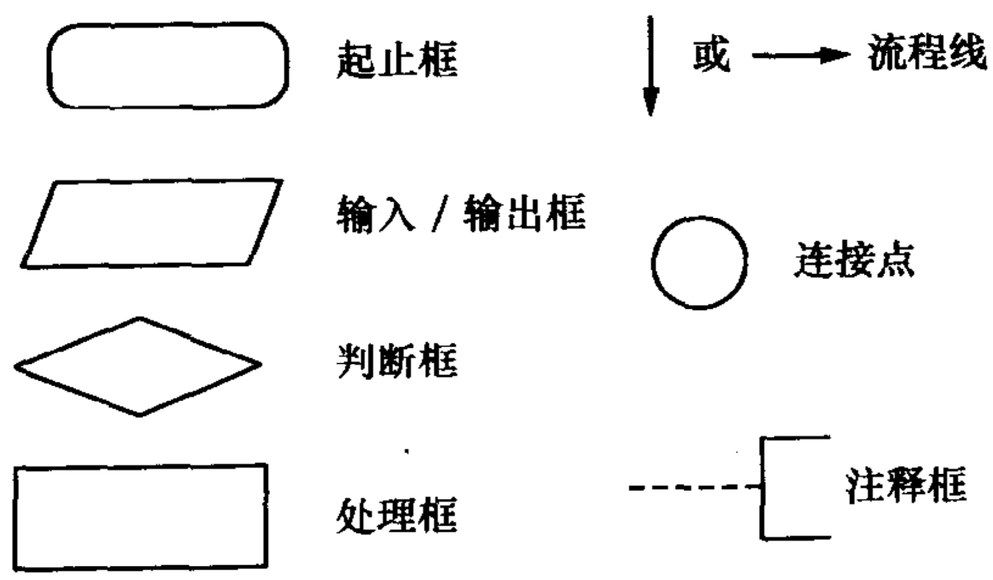

图 1-2　常用的流程图符号

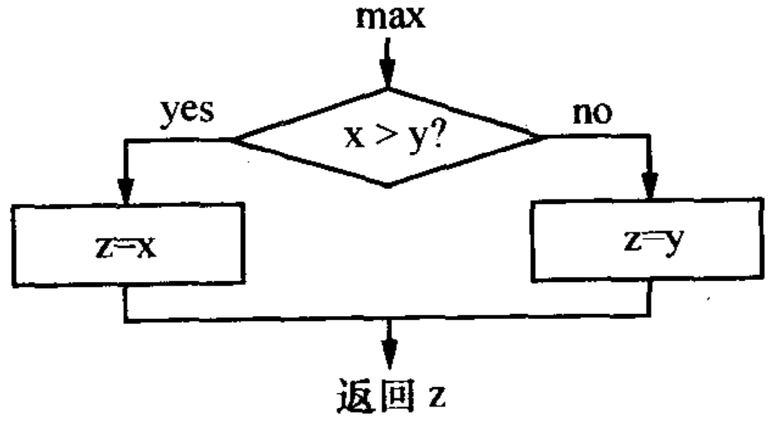

图 1-3　函数 max 的流程图

1.2 C语言的基本词法

任何一种语言，都有自己的单字、单词和语句的构成规则。学会了这些知识，才能用它们书写出精彩的文章。C 语言作为计算机的一种程序设计语言，当然也有它自己可以使用的字符集、基本词类（保留字），也有自己的各种规则和语法。只有学习、遵从它们，才能编写出符合要求的各种程序来。

1.2.1 字符集

允许出现在 C 语言源程序中的所有字符的总体，称为 C 语言的“字符集”。它由数字、英文字母、图形符号以及转义字符 4 部分组成。

（1）数字：10 个十进制的数字，即 1，2，3，4，5，6，7，8，9，0。

（2）英文字母：26 个大写英文字母 A～Z，26 个小写英文字母 a～z。

（3）图形符号：表 1-1 列出了 C 语言允许使用的图形符号。

表 1-1　　C 语言图形符号表

～	波浪号	)	右圆括号	:	冒号
`	重音号	_	下划线号	;	分号
!	惊叹号	—	减号	"	双引号
@	A 圈号	+	加号	'	单引号
#	井号	=	等号	<	小于号
$	美圆号	|	或符号	>	大于号
%	百分号	\	反斜杠号	,	逗号
^	异或号	{	左花括号	.	句号
&	与符号	}	右花括号	?	问号
*	星号	[	左方括号	/	正斜杠号
(	左圆括号	]	右方括号		空格

（4）转义字符：在 C 语言源程序中，可以用在反斜杠号（\）后面跟随特定的单个字符或若干个字符的方法，来表示键盘上的字符以及某些不可见的功能控制符。这时，尾随反斜杠后的字符就失去了原有的含义，而赋予了新的特殊含义。通常称反斜杠号为转义符，称反斜杠以及随后的字符整体为一个“转义字符”。表 1-2 是 C 语言的转义字符表。

表 1-2　　C 语言的转义字符表

转义字符	含　义	转义字符	含　义
\n	回车换行符	\a	响铃符号
\t	Tab 符号	\"	双引号
\v	垂直制表符号	\'	单引号
\b	左退一格符号	\\	反斜杠
\r	回车符号	\ddd	1～3 位 8 进制数 ddd 对应的键盘符号
\f	换页符号	\xhh	1～2 位 16 进制数 hh 对应的键盘符号

注意：只有把转义符（反斜杠）放在表 1-2 里面所列出的字符前面时，才能构成转义字符，否则不起任何作用。比如\w，由于反斜杠后面跟随的字符 w 不在表 1-2 里，所以不构成转义字符，它被视为是小写字母 w。

例 1-4　区别“n”和“\n”。

解：当程序中出现“n”时，代表的是英文中的一个小写字母，比如例 1-2 里的变量 n；当程序中出现“\n”时，反斜杠后跟随的 n 就不再是英文里的小写字母 n，这个整体被视为回车换行符。所以，在例 1-2 或例 1-3 的 printf()中的“\n”，表示回车换行符。

例 1-5　在 C 语言程序中写“\101”、“\x41”，它们分别表示什么意思？

解：按照表 1-2 的规定，在反斜杠后跟随 1～3 位数时，就把这些数字理解为是某个键盘符号所对应的 8 进制 ASCII 码值。101 这个 8 进制数，恰是大写字母“A”的 8 进制 ASCII 码值。所以，“\101”就是大写的英文字母“A”。类似地，应该把“\x41”里的 41 视为键盘符号对应的 16 进制 ASCII 码值。因此，它也是大写的英文字母“A”。

注意：“\xhh”里的字符“x”，只起到一个标识后面的数是 16 进制的作用，没有别的含义。由例 1-5 可知，转义字符“\ddd”、“\xhh”，向用户提供了一种用 8 进制或 16 进制的 ASCII 码值来表示各个字符的方法。

1.2.2　保留字

在 C 语言中，具有特定含义的、用于构成语句成分或作为存储类型和数据类型说明的那些单词，被统称为“保留字”，有时也称为“关键字”。C 语言的保留字只能小写。表 1-3 列出了 C 语言中可以使用的所有保留字。随着学习的深入，这些保留字我们基本上都会遇到的。

表 1-3　　**C 语言的保留字表**

保　留　字	含　　义	保　留　字	含　　义	保　留　字	含　　义
char	字符型	void	空值型	while	当
int	整型	const	常量型	do	做
long	长整型	volatile	可变量型	break	终止
short	短整型	auto	自动	continue	继续
float	单精度实型	extern	外部	goto	转向
double	双精度实型	static	静态	return	返回
unsigned	无符号型	register	寄存器	switch	开关
signed	有符号型	typedef	类型定义	default	缺省
struct	结构式	if	如果	case	情况
union	共用式	else	否则	sizeof	计算字节数
enum	枚举式	for	对于		

1.2.3　标识符及其构成规则

在 C 语言中，用户为了区分程序中出现的常量、变量、函数和数组等，就给它们取不同的名字。组成名字的字符序列，称为“标识符”。因此，标识符是用户给程序中需要辨认的对象所起的名字。一个标识符必须符合下面所列的语法规则。

（1）标识符只能以字母或下划线开头。