高等职业院校计算机教育规划教材
Gaodeng Zhiye Yuanxiao Jisuanji Jiaoyu Guihua Jiaocai

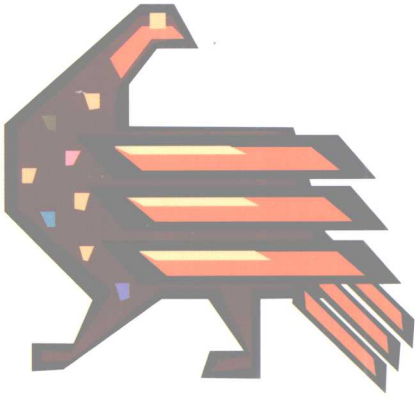
C语言程序设计教程

（第2版）

C YUYAN CHENGXU SHEJI JIAOCHENG

宗大华 陈吉人 编

- 以通俗流畅的语言讲述C语言语法
- 以丰富示例介绍C语言的编程技术
- 以程序运行结果验证C语言语法的各种规定



人民邮电出版社
POSTS & TELECOM PRESS

精品系列

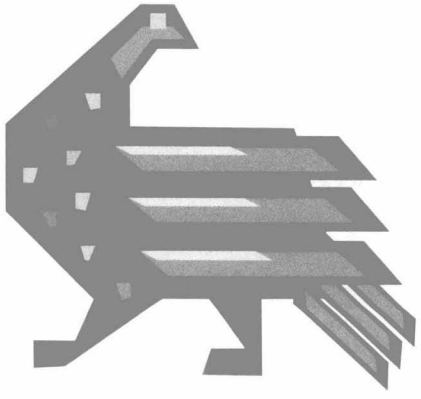
高等职业院校计算机教育规划教材
Gaodeng Zhiye Yuanxiao Jisuanji Jiaoyu Guihua Jiaocai

C语言程序设计教程

（第2版）

C YUYAN CHENGXU SHEJI JIAOCHENG

宗大华 陈吉人 编



人 民 邮 电 出 版 社
北 京

精品系列

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. —2 版. —北京：人民邮电出版社，2008.11（2009.6 重印）
高等职业院校计算机教育规划教材
ISBN 978-7-115-18700-0

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（2008）第 129208 号

内 容 提 要

本书系统讲解 C 语言程序设计的基本知识和方法。内容分为 8 章，包括概述，数据类型、运算符与表达式，3 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。从第 4 章开始，每章都有“程序设计示例”一节，列出 2～3 个较大的程序，力求使学生能够综合运用已学知识，扩大眼界。每章的最后还配有适量的练习题。

本书可作为高职高专计算机及相关专业的教材，也可作为成人教育和职工培训教材。

高等职业院校计算机教育规划教材

C 语言程序设计教程（第 2 版）

◆ 编　　宗大华　陈吉人
责任编辑　李育民

◆ 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
北京世纪雨田印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：18.25
字数：446 千字　　2008 年 11 月第 2 版
印数：20 001 –21 500 册　　2009 年 6 月北京第 2 次印刷

ISBN 978-7-115-18700-0/TP

定价：29.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

丛书出版前言

目前，高职高专教育已经成为我国普通高等教育的重要组成部分。在高职高专教育如火如荼的发展形势下，高职高专教材也百花齐放。根据教育部发布的《关于全面提高高等职业教育教学质量的若干意见》（简称16号文）的文件精神，本着为进一步提高高等教育的教学质量服务的根本目的，同时针对高职高专院校计算机教学思路和方法的不断改革与创新，人民邮电出版社精心策划了这套高质量、实用型的系列教材——“高等职业院校计算机教育规划教材”。

本套教材中的绝大多数品种是我社多年来高职计算机精品教材的积淀，经过了广泛的市场检验，赢得了广大师生的认可。为了适应新的教学要求，紧跟新的技术发展，我社再一次进行了广泛深入的调研，组织上百名教师、专家对原有教材做了认真的分析和研讨，在此基础上重新修订出版。

本套教材中还有一部分品种是首次出版，但其原稿作为讲义也经过教学实践的检验。因此，本套教材集中反映了高职院校近几年来的教学改革成果，是教师们多年来教学经验的总结。本套教材中的每一部作品都特色鲜明，集高质量与实用性为一体。

本套教材的作者都具有丰富的教学和写作经验，思路清晰，文笔流畅；教材内容充分体现了高职高专教学的特点，深入浅出，言简意赅；理论知识以“够用”为度，突出工作过程导向，突出实际技能的培养。

为方便教师授课，本套教材将提供完善的教学服务。读者可到人民邮电出版社教学服务与资源网（www.ptpedu.com.cn）免费下载相关资料。

欢迎广大读者对本套教材的不足之处提出批评和建议！

前言

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，有着极其独特的魅力，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。

本书的第 1 版《C 语言程序设计教程》，是一本专为高职高专学生编写的 C 语言教材。该书针对读者的特点和认知能力，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；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；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。出版以来，该书受到了许多院校师生的欢迎和好评，在此表示由衷的感谢！

在人民邮电出版社编辑的鼓励和支持下，在征集多位授课老师对原书的看法后，对《C 语言程序设计教程》进行了修订。根据多方面的意见和建议，全书在初版取材的基本内容和基本框架下，做了如下几点变动。

（1）删除了原书中的少量内容，对存有的错误做了勘误。

（2）将原书中的大部分例子改为由“程序实现”和“分析与讨论”两个方面来表述。“程序实现”中给出对程序的描述；“分析与讨论”里介绍整个程序的结构，关键变量在程序中起的作用，对程序中难点的解释，或对程序设计另一种方法的讨论等。

（3）从第 4 章开始，在每章的最后都增加了“程序设计示例”一节，列出 2～3 个较大的程序，力求使学生能够综合运用已学的知识，扩大眼界。

本书建议总学时数为 56 学时，授课学时数（不含实践环节）为 44，各章的参考授课学时分配见下表。

章	名　称	授课学时数	章	名　称	授课学时数
第 1 章	概述	3	第 5 章	指针	7
第 2 章	数据类型、运算符与表达式	6	第 6 章	函数	5
第 3 章	C 语言程序设计的 3 种基本结构	8	第 7 章	用户自定义的数据类型	5
第 4 章	数组	5	第 8 章	C 的文件操作函数	5

程序设计即意味着实践！希望读者在学习 C 语言程序设计这门课程时，做到 3 个一定：一定要以极大的热情去动手编写程序；一定要在实践过程中有百折不挠的精神；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，进而才能不断赢取胜利，从而才有可能达到成功的顶峰。

与本书第 1 版配套出版的《C 语言程序设计教程习题解答与实训》一书，给出了本书所有的习题解答，另外，还介绍了 Turbo C2.0 的集成开发环境和调试 C 语言程序的方法，并为各章设计了相应的实训。它是一本配套辅导用书，是对本书内容的一个补充和完善。

本书的编写与修订，得到了多位同事、朋友的支持和帮助。在整个编写过程中，蒋玮、宗杰、黄芳、梁发寅、沈寄云、江汇、余楠等为每章内容的编写、所附习题的收集和调试做了大量的工作，在此不一一详列，仅致以诚挚的谢意。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

2008 年 8 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。这一章就先从外表来看一下 C 语言，以便能在人们的脑海里对它形成一个初步的印象：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，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，对于学习、掌握 C 语言，肯定是至关重要的。

本章着重讲述以下 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 4 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”是指一件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。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（比如 1.2.2 中的保留字）；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（比如第 2 章中的运算符和不等式）。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，计算机也不能直接识别与接受，也必须要有一个“翻译”，先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：一是事先编好一个称为“编译程序”的机器指令程序，它把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。另一个是事先编好一个称为“解释程序”的机器指令程序，它把源程序逐句翻译，译一句就执行一句，这种翻译过程如图 1-1（b）所示。前一种翻译称为“编译”方式，后一种称为“解释”方式。

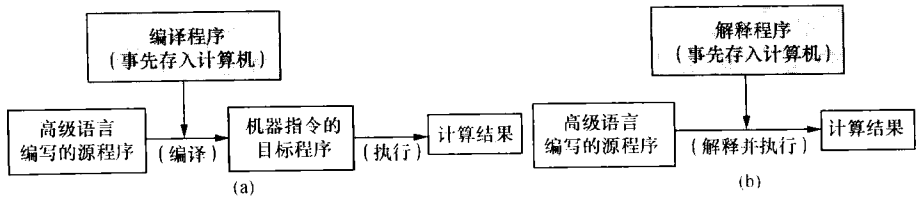


图 1-1　编译和解释两种翻译方式示意图

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的。

从解答中可以看出，机器语言程序完全没有直观性可言，如果不了解机器指令 A1 是表示将跟随其后单元中的内容送至寄存器 AX，那么根本无法知道它的含义。对于汇编语言，MOV 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 中去。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，它简直就近乎于是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 语言来编写程序时的通常做法，从而归纳出 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　（1）程序实现

```
#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 */
}
```

（2）分析与讨论

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（即单词 integer 的缩写）。

第 2 条语句“scanf ("%d%d", &m, &n);”是一条格式输入语句，其中的&m 和&n 表示变量 m 和 n 所对应的内存单元地址。其功能是按照格式符“%d”的规定，从键盘接收两个十进制的输入数据（之所以是十进制，是由格式符%d 中的字母 d 限定的），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变量的一个

操作过程。

第 4 条语句“printf ("sum=%d\n", sum);”，是一条格式打印输出语句，表示将变量 sum 的当前值，按照格式符“%d”的规定输出一个十进制整数。比如说，如果现在从键盘上输入的两个数是 3 和 5，那么，在显示器上就应该输出信息：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，将其中的大数打印输出。

解 （1）程序实现

```
#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

（2）分析与讨论

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（注意，程序中的箭头线是用来表明函数间的调用关系的）。

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，把大数存入变量 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 中的结果值送给函数 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，然后才能真正动手去编写程序和上机调试。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；可以借助于所谓的 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。

本书列举的程序都是比较简单的，因此不去专门关注算法的描述。不过为了帮助对所编程序的理解，有时会给出程序的流程框图。图 1-2 给出了画流程框图时常用的一些符号。

比如可以利用这些符号，为例 1-3 中“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。

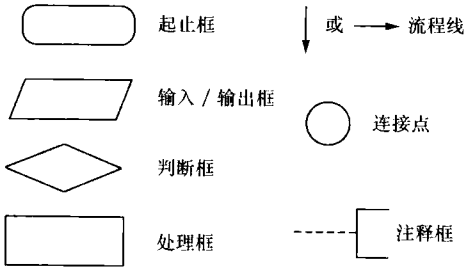


图 1-2　常用的流程图符号

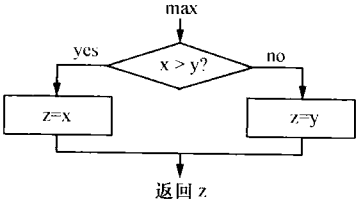


图 1-3　函数 max 的流程图

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，当然也有它自己允许使用的字符集、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

出符合要求的各种精彩程序来。

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（比如退格、换行等）。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 中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 中，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 就不再是英文中的小写字母 n，这个整体被视为是回车换行符。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 进制数相当于十进制数 65，查书后的附录 2，知道是大写字母“A”。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”中的字符“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。

（2）在第一个符号的后面，可以跟随字母、数字或下划线。

（3）标识符中区分字母的大、小写。

（4）标识符的长度一般不超过 8 个字符。

（5）C 语言的保留字不能作为标识符使用。

例 1-6 试判断下面所给出的字符序列，哪一个是正确的 C 语言标识符。

x　　_906　　A203　　aBBC　　C.508　　int

y-56　　gb?　　b_B64　　2abc　　ABBC

解 根据构成标识符的语法规则可知，上述字符序列里，正确的标识符是：

x　　_906　　A203　　aBBC　　b_B64　　ABBC

不正确的标识符是：

C.508（句号不允许出现在标识符里）　　y-56（减号不允许出现在标识符里）

gb?（问号不允许出现在标识符里）　　2abc（标识符不允许以数字开头）

int（保留字不允许作为标识符）

注意：由于 C 语言区分大、小写，所以 aBBC 和 ABBC 是两个不同的标识符。

1.3 Turbo C 2.0 开发环境简介

真正要运行一个用 C 语言编写的程序，至少要做如下的 4 项工作。

（1）编辑。通过使用编辑器，把 C 语言程序录入计算机，并以文件的形式存放到磁盘上，这个过程称为“编辑”。它将产生出以“.C”为扩展名的源程序文件。

（2）编译。源程序不能直接执行，必须通过 C 编译程序（本书使用的 C 编译程序是 Turbo C 2.0 版本）将它“翻译”成由机器指令组成的目标程序，这个过程称为“编译”。它产生出以“.OBJ”为扩展名的目标程序文件。

（3）连接装配。目标程序仍不能立即在机器上执行，因为程序中还会用到 C 语言自身提供的系统库函数，例如 printf()、scanf()等，需要把它们与产生的目标程序连接在一起，形成一个整体，这个过程称为“连接装配”。它将产生出以“.EXE”为扩展名的可执行程序文件。细心的读者在阅读例 1-2、例 1-3 时会发现，在它们的最前面有：

```
#include "stdio.h"
```

这是一条编译预处理命令（将在第 7 章介绍），它的作用是告诉编译程序：在这个程序中，用到了哪个范围内的系统库函数，以便编译程序能够找到它们，并把它们与所编写的源程序相连接。由于这两个例中都用到 printf()、scanf()，它们是 C 语言提供的输入输出函数，所以在编写的程序最前面，要加上这条编译预处理命令。

（4）执行。运行可执行文件，以获取所需要的结果。

这 4 项工作的整个流程如图 1-4 所示。

Turbo C 2.0 向使用者提供的是一个集成开发环境。也就是说，在 Turbo C 2.0 所提供的环境下，用户可以完成编辑、编译、连接装配以及运行的所有工作。下面，对 Turbo C 2.0 的使用做一个简要的介绍。