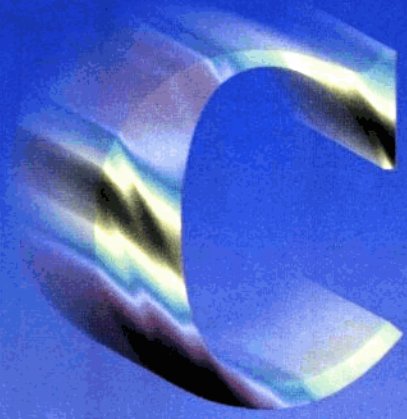


C语言程序设计教程

郑国平　胡海明　编著

冶金工业出版社

前　言

C语言是一种具有良好的移植性和灵活性的计算机软件程序设计语言，是近年来国内外推广使用最为迅速的一种现代编译型程序设计语言。使用C语言进行程序设计已成为软件开发的一个主流。由C语言编写的各种程序，已经广泛应用于人们生产、生活的各个领域。C语言已成为广大计算机工作者必须掌握的重要工具之一，也是高校大学生亟待学习的重要内容。

在多年的C语言教学工作中，作者深感教材的重要性，因而萌发了编写此书的念头，意在向广大大专院校学生和计算机工作者呈献一本内容全面、讲解详尽、深入浅出、易学易会的C语言程序设计书籍。

本书详细讲解了C语言程序设计概念及上机步骤；基本数据类型及表达式，分支、循环及结构化程序设计的基本思想；数组、指针及其相互关系，函数及其在模块化程序设计中的体现；结构体、共用体及用户自定义类型的使用，文件的基本操作及其应用。每章均附有大量习题，分为选择题、填空题、编程题、上机练习题4种类型，以帮助学习者消化吸收每章知识。

本书共分7章，由郑国平主编，李畅主审。其中第2、3、6章及附录由郑国平编写；第1、4、5、7章由胡海明编写。全书由郑国平负责修改并定稿。由于作者受经验和水平所限，书中疏漏不当之处在所难免，敬请各位专家、读者批评指正。

编　者

1998年8月

目　录

1 C语言入门

1.1 计算机语言及程序设计

1.1.1 计算机语言的概念及发展史

人们的日常生活总离不开语言。语言可以表达我们的思想、感情，传递一定的信息。这种语言（如我们的母语）是在人类漫长发展过程中逐步成熟起来的。语言，特别是西方语言有着严谨的语法，但是这些语法是后人对语言使用的一种归纳总结。它们作为一种规范，是逐步成熟起来的，有其自然的发展过程。从这种意义上说，它们是自然语言。

然而，随着人类社会的发展，以及人类社会的科学研究及生产、生活应用领域的不断拓展，自然语言已经不能满足人们的需要，于是人们发明了一些人工语言。之所以称其为人工语言，是因为人们先设定规则、约定，然后再在这些规则、约定的基础上表达自己的思想，传送一定的信息进而完成特定的功能。在某些领域中，它起到了自然语言所无法替代的作用。例如，在通讯领域，人类的自然语言在空气媒介中只能传输极短的距离，借助于莫尔斯码这种人工语言，就能把它传输到很远的地方。

由于计算机的出现，人们总想用它们处理一些数据。此时，很自然就产生了这样一些问题：这些数据如何表示呢？如何进行各种运算呢？从早期的计算机语言到我们现在所学的C语言，已经经历了三个发展阶段。开始，为了便于机器接受，人们直接用0 1组成的字符来表示数据以及数据之间的运算，比如用11000110表示把某个数和另一个数相加。这是第一代计算机语言，称之为机器语言。显然，它使用起来很不方便，于是人们发明了助记符，对每一步骤用类似于自然语言的符号进行描述，比如以上两数相加的过程被表述为ADD A,B。这是第二代计算机语言，称之为汇编语言。由于汇编语言是直接面向机器的，对硬件系统的依赖性比较大，同时这种表述与人们通常的表述方式差别还比较大，于是人们越来越强烈地希望能使用一种适用程序设计的语言。迫于此种情况，人们不断发明了一些通用程序设计语言。它们不仅可以用自然语言表述，而且可以像自然语言那样表述解决问题的过程，比如以上两数相加的过程直接描述为a+b，不再考虑其实现的细节。C语言正是这样一种通用程序设计语言。

1.1.2 程序设计的概念及过程

如前所述，人们运用计算机是要来解决一些问题的，而问题的解决总有一定的程式或步骤，比如要完成“看电视”这一任务，必须：(1)坐到电视前；(2)打开电视机；(3)调到所喜欢的频道；(4)欣赏节目。显然，用计算机来模拟这一过程，当然也要反映这些步骤。这种用计算机程序设计语言表述的、反映出问题解决过程的代码集就称为程序。从抽象的角度看，解决问题总有一定的对象和规则，对象间的相互作用乃至关联过程也必须得到描述，这两个方面就构成了程序的基本内容。程序设计是指设计、编制、调试程序的全过程。与做任何事一样，

程序设计时首先要考虑的就是设计的目的，进而考虑必须实现的各种功能(即确定做什么)，这是设计阶段的任务。接着，必须编写出能实现上述功能的程序代码(即确定如何做)。最后，必须进行上机调试，消除存在于程序代码中的各种错误，进而确认目的的最终实现。从方法论上看，程序设计有各种各样的策略，有结构化程序设计、面向对象程序设计、事件驱动的程序设计等。C 语言适合结构化程序设计方法的实现。

1.2 C 语言的由来及特点

1.2.1 C 语言的由来

C 语言的产生有个曲折的过程。随着通用程序设计语言(亦称高级语言)的产生，人们对它们的研究越来越深入，人们已经不满足于一般的应用程序的编写，要求系统程序也用高级语言来描述，比如对操作系统进行描述。但是，一般的高级语言难以实现汇编语言的许多功能。鉴于此，人们逐渐开发出了既有一般高级语言的功能又具较强的低级语言(即汇编语言)功能的语言——C 语言。为了实现这种融合，开始人们推出了混合编程语言(combined program)，简称 CP 语言。由于它规模庞大，难以实现，人们抽出其内核，构造了基本混合编程语言(BCPL)，乃至更加简约的 B 语言。由于 B 语言过于简约，1973 年，贝尔实验室的研究人员在 B 语言基础上开发出了能描述 unix 操作系统的 C 语言，此处的 C 是 combined 一词的缩写，显然它强调了其混合性，这是我们在学习、应用 C 语言过程中必须注意的。

C 语言自产生后的发展过程是，如前所述，开始出现的 C 语言是 C 编译语言；1983 年出现了 ANSIC(它是由美国国家标准协会即 ANSIC 制定的标准)；1987 年又推出了 87ANSI _ C，本书描述的就是这种新标准。

1.2.2 C 语言的特点

C 语言是目前最为流行的几种程序设计语言之一，它的流行得益于其具有以下明显优于其他程序设计语言的特性：

(1)简洁、紧凑、表达灵活方便。比如，在 pascal 语言中，用 begin—end 标识一个程序段的开始与结束，而 C 语言简洁地记为{}。当然，C 语言的简洁主要体现在其代码的简约，比如用 a＋＋取代一般语言的 a＝a＋1，代码的简洁使其拥有很高的编译效率。

(2)支持结构化程序设计。结构化程序设计的核心是自顶向下，逐步细化的编程策略，进而形成以若干关键数据结构为核心的、形状类似于倒置的枝繁叶茂的树状模块结构。C 语言基本上可以说是函数式语言，这使得它很方便地实现这种构造。

(3)运算符丰富。C 语言不仅拥有一般高级语言都有的运算符，而且还拥有不少独特的运算符，如赋值运算符、自增自减运算符、位运算符。其中的位运算符能使其直接对硬件进行操作。

(4)数据类型也特别丰富。C 语言拥有整型、实型、字符型、数组、指针、结构体、联合体、枚举、用户自定义数据类型，能方便地实现各种复杂的数据结构，有很强的数据处理能力。

(5)较强的编译预处理功能。C 语言能用＃define 、＃include 等编译预处理语句，实现各种宏定义，实现对外部文件的读取、合并，这就为开发较大规模的程序提供了很大的方便。另一方面，C 语言还具有条件编译功能，即分块编译调试，最终再联合编译、运行。这显然大大提高了程序开发的效率。

C语言的双重特性——既有高级语言的特性又能实现汇编语言的大部分功能，使得人们对C语言的掌握比其他语言稍难一点，但是，人们一旦掌握它，便会拥有很强的表达力，能编写任何类型的程序。从另一方面看，任何事物都是辩证的。由于历史条件的限制，传统的C语言也有其自身无法克服的困难。它基本上是面向数据和过程的，整个展开过程和数据组织是分离的，数据和程序始终有不相匹配的可能。由此，人们设计了C++语言，它吸收了C语言的一切优点，同时又克服了以上困难，实现了对数据的封装，引进了许多新的机制，进而实现了更高一个层次的程序设计方法——面向对象的程序设计，它为C语言跨入21世纪展现了一个无限美好的前景。

1.3 一个简单的C语言程序

以下就是一个简单的C语言程序。由此，我们能够得到对C语言程序的一个最简单也是最基本的认识。

例[1.1] 求圆柱体的体积，其中各主要符号的含义如下：

v 表示所求的体积；r 为圆柱的半径；h 为圆柱体的高。

能实现这一功能的程序代码如下(其中的代码及其实现的细节在以后的章节中再作介绍)：

```
#define PI 3.1415926                          /*行1/
main( )                                       /*行2*/
{ int v,h,r;printf("input r,h");              /*行3*/
  scanf("%d%d",&r,&h);                        /*行4*/
  v=c(r,h);                                   /*行5*/
  printf("v=%d\n",v);                         /*行6*/
}
   int c(x,y)                                 /*行7*/
   int x,y;                                   /*行8*/
  {int z; z=PI*x*x*y;                         /*行9*/
  return(z);}                                 /*行10*/
```

这个程序由两个函数组成，即第2行的主函数main( )以及第8行的c(x,y)。main( )是程序主控函数的标识，c(x,y)是被它调用的函数。

第2行是对函数内用到的数据(v,h,r)的类型说明。

第3行是数据输入的提示。

第4行是数据输入(具体数值由运行时输入)。

第5行是调用函数c( )。

第6行是数据输出。

第6行后的右大括号，与第2行后的左大括号配对，标识这是个完整的程序块。

从以上的分析看，C语言程序一般由若干函数组成，每个函数内部大体包括函数说明与函数体两部分；从另一角度看，程序又由若干行组成，每行又由若干语句(以分号标识)组成，而程序(或程序段)则由一对大括号标识。

仔细观察以上的程序，可以看出 C 语言程序的书写格式有以下几个特点：

(1)C 语言程序一般用小写英文字母书写，常量的宏定义和其他特殊的符号才用大写字母。

(2)书写自由，一行中可以书写多个语句，一个语句也可以占多行，不同语句之间必须以分号分隔开。

(3)不同结构层次的语句，一般缩进一些位置，同一结构层次最好上下对齐。

(4)所有程序(或程序段)以大括号为标识。

以上是对 C 语言程序形式特征的分析。如果从程序的逻辑结构看，它一般由以下 3 部分组成：

(1)输入部分。如程序的第 4 行。

(2)处理部分。如程序中的第 5 行。

(3)输出部分，如程序中的第 6 行。

这是我们必须牢记的内容，因为在以后的学习中，我们会逐步看到，编程过程一般都是依照这个思路展开的。

1.4 C 语言程序的开发过程

C 语言程序的开发过程一般是依照图 1.1 所示的过程展开的。

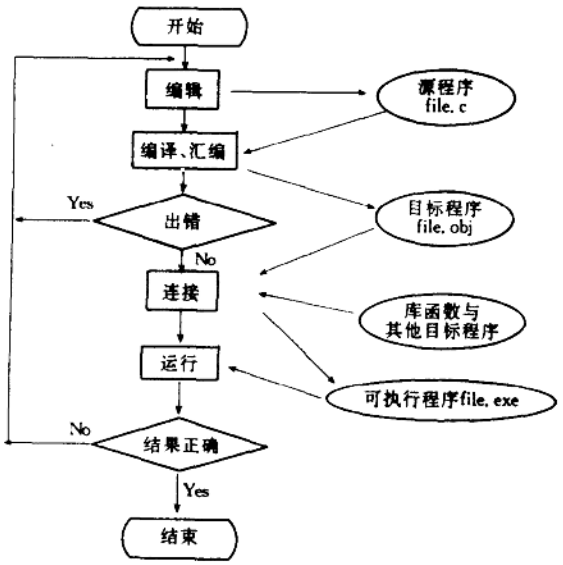


图 1.1 C 语言程序的开发过程

从图 1.1 可以看出，C 语言程序开发的第一步是编辑。通过键盘向计算机屏幕录入源程序代码的过程，需要系统提供的屏幕编辑程序支持；借助于它，错误代码才能被修正。若认为输入无误，就应该对源程序进行编译，即检查其词法、语法、语义方面是否存在错误；通俗点说，就是检查是否有拼错的关键词，是否有不合 C 语言语法规则的表达方式以及语义方面

的矛盾或含混。这些错误在编译阶段都可以查出。如图 1.1 所示，在发现错误后，反复修改，编译成功后，就生成了目标程序，它与库函数乃至其他目标程序一起装配连接形成可执行文件(倘若装配连接成功)。然后运行该程序就可以得到一定的输出结果。若输出结果完全正确，开发过程就结束了；倘若不正确，必须再回到编辑阶段，重新检查可能导致错误结果的源程序代码。必须指出的是，此时解决的是逻辑上的错误，而编译阶段解决的基本上是词法、语法错。若修改逻辑错误，在编译、连接阶段还是可以发现一些线索的，因为此时计算机屏幕给出了许多警告，这些警告就告诉用户在程序中还存在哪些潜在的可能导致错误结果的问题。一般说来，用户从中还是可以分析出产生错误运算结果的症结所在，进而改正错误，得到正确的运行结果。

综上所述，C 语言程序开发基本上可以分为 4 个阶段：

(1)编辑。此时借助于系统提供的编辑程序(一般是全屏幕编辑程序)，输入事先编写的源程序，源程序文件以文本文件的方式存放在外存，文件名由用户定，其扩展名均为 .c。

(2)编译。借助于系统提供的编译程序，对源文件进行编译，检查其词法、语法、语义方面的错误。若一切正确，产生由目标代码组成的目标文件(以.obj 为后缀)。

(3)连接目标文件及库文件。编译产生的目标文件一般不能直接运行，还必须把所引用的函数以及源程序指定的一些目标文件连接起来，形成完整的可执行文件。

(4)运行程序。连接成功后，即可运行，此时用户只需键入源程序的文件名即可。

1.5 TC 环境下的上机步骤

1.5.1 TC 环境简介

TC 即 TURBO C 是一个集成的 C 程序开发环境，其主菜单有七个选项：

(1)File，它包括文件的装入(load)、保存(save)换名(write to)等功能。

(2)Edit 建立、编辑源文件

(3)Run 自动编译、链接并运行程序。

(4)Compile 将源程序编译并装配为目标文件。它拥有编译成.obj 文件，生成连接 .exe 文件等功能。

(5)Project 处理工程，它包括工程名、打断编译、清除工程等功能。

(6)Options 选择编译关系，它包括存储模式、宏定义、代码生成、优化，还包括连接程序、环境、目录等功能。

(7)Debug 跟踪排错。

1.5.2 TC 环境下的上机步骤

(1)先进入 TC 环境。此时，需在磁盘的根目录或子目录中找到 TC，键入 CD\TC 再键入 TC 即可。

(2)利用 File(按功能键 F3 即可)菜单中的 load 项，键入所需的文件名，然后通过键盘输入编好的源程序。

(3)利用 Compile 项(按功能键 F9 即可)编译源程序，检查、排除程序中存在的错误。

(4)利用 Run 项(按功能键 F10，再把光标移至'run'处，按回车键即可运行编译成功的目标程序。若需要输入数据(最好有输入提示)，必须按程序要求的格式键入数据。

(5)存盘。无论完成到哪一步,倘若以后仍需要修改、查看或使用它,用 F2 存盘。

依照上述步骤,完全可以在 TC 环境中运行一个 C 程序。但是需要指出的是,编译过程中,改正错误是个比较繁琐的过程,需要很大的耐心与细致。若有错,一般在屏幕的下半部分会显示出错的行号及其原因,按任意键后,光标会停在出错行。根据屏幕提示,许多词法、语法方面的错误还能排除;至于运行中出现的错误,修改起来比较复杂。

习 题 一

一、填空题

1. 所谓程序是指_____。
2. 所谓程序设计是指_____。
3. C 语言的特点是_____。
4. C 语言的上机步骤_____。

二、上机练习

1. 练习 TC 环境下上机的步骤。

```
main( )
{
  printf("wish you success");
}
```

2. 请参照例 1,编写 C 语言程序,显示以下信息:

```
# # # # # # # # # # # # #
Wonderful ! Let's try !
# # # # # # # # # # # # #
```

2 程序设计基础

计算机的基本功能是数据处理。数据处理的基本对象是常量和变量。C 语言提供了多种数据类型的常量和变量。常量、变量和运算符结合形成表达式。表达式是 C 语言的最基本要素。C 语言运算符种类十分丰富，包括算术运算符、关系运算符、逻辑运算符、位操作运算符以及某些特殊的运算符，具有丰富的运算功能。本章介绍 C 语言的基本数据类型及其对应的常量和变量，以及类型之间的转换，介绍算术运算、赋值运算和某些特殊运算，此外还将简单介绍 C 程序中经常使用的输入输出函数。

2.1 标 识 符

直接地说，标识符就是一个名字，我们以后学习的符号常量名、变量名、函数名、标号、数组名、文件名、结构类型名和其他各种用户定义的对象名都是标识符，它们的命名必须满足标识符的构成规则。

2.1.1 标识符的组成

C 语言允许用作标识符的字符有：

(1)26 个英文字母，包括大小写(共 52 个)；

(2)数字 0,1,…,9；

(3)下划线_。

2.1.2 标识符的构成规则

标识符的构成规则为：

(1)必须由字母(a～z,A～Z)或下划线(_)开头；

(2)后面可以跟随任意的字母、数字或下划线(_)；

(3)对 Turbo C 较高版本而言，标识符的有效长度是 31 个字符；前 31 个字符有效，后面字符无效。

2.1.3 注意事项

对标识符的注意事项为：

(1) 在 C 语言中，大小写字母有不同的含义，例如：num ，Num ，NUM 为三个不同的标识符。

(2) 在构造标识符时，应注意做到“见名知意”，即选有含意的英文单词(或汉语拼音)作标识符，以增加程序的可读性。如表示年可以用 year，表示长度可用 length，表示和可以用 sum 等。

(3) C 语言中有一些标识符被称为关键字，在系统中具有特殊用途，不能作为一般标识符使用，如用于整型变量说明的 int 关键字，就不能再用作变量名。

(4) 有些标识符虽不是关键字，但 C 语言总是以固定的形式用于专门的地方，因此，用

户也不要把它们当做一般标识符使用，以免造成混乱。这些标识符常用的有：

define include ifdef ifndef endif elif

合法的标识符示例：sum，a1，i，j5k3，sum _ ave

不合法的标识符示例：5i 错在以数字开头；

u.s 错在出现"."；

good bay 错在中间有空格。

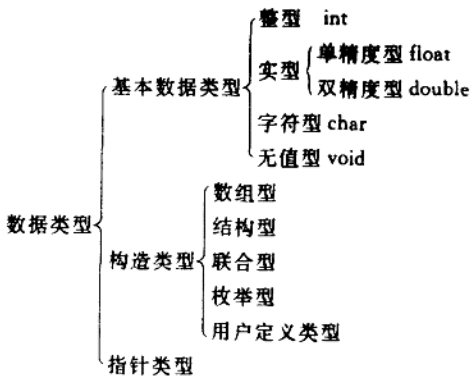
2.2 C 语言的数据类型

C 语言提供了丰富的数据类型。C 语言的数据类型有基本数据类型、构造类型和指针类型。

基本数据类型包括：整型、字符型、实型（又称浮点型）、无值型。实型又包括：单精度型和双精度型。

构造类型包括：数组、结构、联合、枚举、用户定义类型。

综合表示如下：



本章介绍基本数据类型。构造类型和指针类型将在以后的章节里分别介绍。

2.3 常 量

常量是指在程序运行过程中其值不发生变量的量。C 语言中的常量有：整型常量、实型常量、字符常量和字符串常量。

2.3.1 整型常量

整型常量简称整数。C 语言中有三种形式的整型常量。

A 十进制整数

十进制整数即通常习惯的十进制整数形式，如：

10 −345 0 +5 123

B 八进制整数

八进制整数是以 0 开头的八进制数字串，只能使用数符 0～7。下面是一些合法的八进

制整数：

0123 （表示十进 83）

0400 （表示十进 256）

0177777(表示十进 65535)

C 十六进制整数

十六进制整数是以 0x 或 0X 开头的十六进制数符串。十六进制数符有 0～9,a～f(或 A～F),其中 a～f(或 A～F)分别表示十进制值 10～15。下面是一些合法的十六进制整数。

0X123 （表示十进制整数 291）

0X12 （表示十进制整数 18）

0XFFFF （表示十进制整数 65535）

八进制、十六进制整数没有符号,即没有负的八进制、十六进制数。

C 语言中,整数取值范围一般由 CPU 所处理的机器字的位数所决定。一个整数以两个字节存储:整数用十进制数表示的范围为－32768～32767;用八进制表示的范围 0～0177777;用十六进制表示的范围为 0X0～0XFFFF;无符号整数值的范围是十进制 0～65535。数值超过范围时,系统自动按长整数进行存储。长整数每个数值占 4 个字节(32 位)存储空间,长整数的表示形式是在数字的后面跟上字母 L 或 l,如:

12L －100L 1234567L 0200000L 0X12L 0X10000L

看起来,12L 和 12 好象没有什么差别,但在用 16 位二进制表示整数的计算机上,12 占 2 个字节存储空间,而 12L 占据 4 个字节。

十进制长整数的取值范围是:－2147483648～2147483647

2.3.2 实型常量

实型常量也称浮点型常量,即实数。C 语言中的实数有两种表示形式。

A 一般形式

一般形式的实数由数字、小数点和可能的正负号组成,如:

0.373 .365 365. 365.0 0.0 －3.65

B 指数形式

指数形式的实数由尾数部分、字母 e 或 E 和指数部分组成。例如:

345E2 表示实数 345×10^2

－28.127E8 表示实数 -28.127×10^8

实型常量不管表现形式如何,总是占据 8 个字节的存储空间(即以双精度的形式出现)。

[例 2.1] 分别输出常量 12,1234567,12L,12.0,12E2 在内存中存储所占用的字节数。

```
main( )
  {
  printf("%d\t",sizeof(12));
  printf("%d\t",sizeof(1234567));
  printf("%d\t",sizeof(12L));
  printf("%d\t",sizeof(12.0));
  printf("%d\t",sizeof(012E2));
```

结果是

2 4 4 8 8

程序的输出结果表示 12,1234567,12L,12.0,12E2 在计算机内存中存放分别占用 2、4、4、8、8 字节。

2.3.3 字符型常量

字符型常量是用单引号括起来的一个字符,如:

'a' 'A' '0' '*'

C 语言的字符常量占据一个字节的存储空间,在存储单元中存放的并不是字符本身,对大多数系统而言,存放是字符对应的 ASCII 码,见附录一。在这种情况下,存储字符'a'的单元实际值是 97,存放'0'的单元实际值是 48。

'a'→97 '0'→48

由于字符常量说到底是一个整数,因此它可以像整数一样参加数值运算。

例如:

a='A'+6;等价于 i=65+6;

y='?'+'B';等价于 j=63+66;

当然,字符常量的主要用途是用于字符之间的比较。

除了以上形式的字符常量外,C 语言还允许使用一种以特殊形式出现的字符常量,就是以"\"开头的转义字符。转义字符常常用于表示 ASCII 字符集内的控制代码和某些用于功能定义的字符,如:

ASCII 字符集中的控制代码 X0d 表示回车,C 语言中使用转义字符'\r'表示。即'\r'实际上是一个字符,它的 ASCII 为 X0d。常见的以\开头的转义字符,见表 2.1。

表 2.1 转义字符

转义字符	意义	ASCII 码
\a	响铃	0X07
\n	换行	0X0a
\t	水平 tab	0X09
\v	垂直 tab	0X0b
\b	退格(Back space)	0X08
\r	回车	0X0d
\f	换页(走纸)	0X0c
\0	空字符(null)	0X00
\\	反斜线	0X5c
\'	单引号'	0X27
\"	双引号""	0X22
\ddd	1~3 位八进制所代表的字符	对应字符的 ASCII

字符常量中单引号不能表示成“'”,而应表示为“\'”。

反斜杠字符常量表示为“\\”;

回车换行字符常量“\n”;

双引号字符常量“\"”;

转义字符\ddd(d 为八进制数字,0～7 之一),它将字符的 ASCII 码值转换为对应的字符,用它表示任一个字符。例如:

\101 表示字符 A　　'\101'表示'A'

\012 表示转义字符\n　　'\012'表示'\n'

2.3.4 字符串常量

字符串常量是用一对双引号括起来的零个或多个字符的序列,如:

"This is a string"

"5401349"

"$ 10000.00"

" "　　引号中有一个空格

""　　引号中什么也没有

"\"　　引号中有一个转义字符

字符串常量在内存中存储时,系统自动在每个字符串常量的尾部加一个“字符串结束标志”字符\0(\0 的 ASCII 码为 0)。因此,长度为 n 个字符的字符串常量,在内存中要占用 n+1 个字节的空间。

例如 "hello"在内存中的形式是:

104	101	108	108	111	0

为了能直观理解,以后表示字符时,直接用字符本身表示。

上例表示成:

h	e	l	l	o	\0

了解了这一点,就可以理解字符串常量"a"和字符常量'a'的区别了。

'a'的存储需一个字节:

A

而"a"的存储需二个字节:

A	\0

""表示一个空字符串,也需占一个字节的存储:

\0

2.3.5 符号常量

常量也可以用标识符来命名。代替常量用的标识符称为符号常量。通常习惯用大写字母表示符号常量,用小写字母表示变量名、数组名等,以便区别。符号常量在使用之前必须预先定义,其定义的格式为:

#define　符号常量名　常量

注意不要以分号结束，它不是语句。

例如：

```
#define  NULL  0
#define  PI  3.1415926
```

[例 2.2] 按输入的半径求圆的周长和面积。

```
#include <stdio.h>
#define  PI  3.1415926
main( )
  {float r,s,l;
  scanf("%f",&r);
  l=2*PI*r;s=PI*r*r;
  printf("r=%6.2fs=%6.2fl=%6.2f\n",r,s,l);
  }
```

运行结果：

```
2.0 ↙
r=3.00s=28.27l=18.85
```

程序第一行：#define PI 3.1415926

定义了一个符号常量 PI，这样在以后的程序中标识符 PI 就可以代替常量 3.1415926 被引用，但在程序中不允许修改它的值。用符号常量代替常量，至少有两个好处：

第一，使用符号常量，可以使程序更易读。例如前面定义的 EOF，它的意思是“文件尾标”，NULL 的意思是字符串尾标，PI 是 π 的谐音，意思是圆周率（根据标识符构造规则，不能用 π 直接作为圆周率符号常量）。

第二，程序更易修改。有些常量，在调试、扩充或移植时要求修改其值，这种常量就应被定义成符号常量。当需要修改常量时，只需改变其定义即可，这样可避免在程序中多处修改，可避免因修改遗漏而导致错误。

2.4 变　　量

所谓变量是在程序执行过程中其值可发生变化的量。

变量必须用标识符来取名，称之为变量名。每个变量都与一个数据类型相联系，类型决定了变量占用的存储空间大小，也就决定了变量可以取值的范围和它可以参加的运算。C 语言规定，程序中的变量在使用之前必须加以说明。变量说明可以在函数前面，也可在函数参数说明处，还可在复合语句说明部分。

变量说明的形式为：

数据类型名　变量名，　变量名，　…，　变量名；

```
例：    int i;          /*说明 i 为整型变量*/
        float f;        /*说明 f 为浮点变量*/
```

在变量说明的同时，也为变量分配存储单元。上例定义 i 和 f 的同时，为 i 分配 2 个字