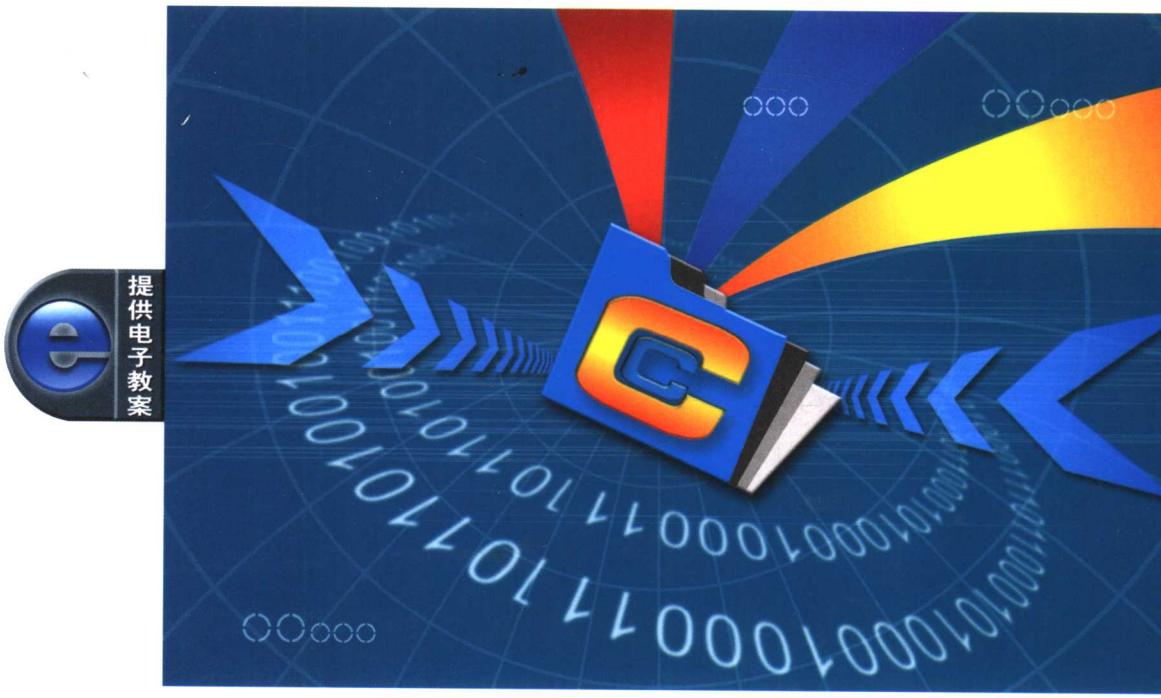
21世纪高等院校应用型规划教材

C语言程序设计



熊 壮 编著

机械工业出版社
CHINA MACHINE PRESS

C语言程序设计

[redacted]

21世纪高等院校应用型规划教材

C语言程序设计

熊 壮 编著

机 械 工 业 出 版 社

本书从结构化程序设计技术的角度出发，以C程序设计语言为载体，通过对C语言的基本语法、语义的讲解以及对各种典型问题的分析，展现了在计算机应用过程中如何将方法和编码相联系的具体程序设计过程，进而向读者介绍计算机结构化程序设计的基本概念、基本技术和方法。

本书选用Visual C++ 6.0作为教学环境，书中的所有教学示例、习题的参考解答都在Visual C++ 6.0集成开发环境中通过测试。

本书可供高等院校计算机及其相关专业计算机技术基础课程教材，也可供计算机应用开发人员参考。

图书在版编目（CIP）数据

C语言程序设计/熊壮编著. —北京：机械工业出版社，2006.12
（21世纪高等院校应用型规划教材）
ISBN 7-111-20406-9
Ⅰ.C... Ⅱ.熊... Ⅲ.C语言－程序设计－高等学校－教材
Ⅳ.TP312

中国版本图书馆CIP数据核字（2006）第139927号

机械工业出版社（北京市百万庄大街22号　邮政编码100037）
策　　划：胡毓坚
责任编辑：李馨馨
责任印制：洪汉军
北京汇林印务有限公司印刷
2007年1月第1版·第1次印刷
184mm×260mm·22.5印张·554千字
0001— 5000册
定价：31.00元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639　88379641　88379643
编辑热线电话：（010）88379739

出版说明

进入信息时代，我国高等教育面临的情况发生了巨大变化。信息技术日新月异，使得与其相关的课程知识结构更新迅速。由于社会对应用型人才的需求日趋强烈，高校也越来越注重对学生实践能力的培养。大多数高校的上机环境、教师的业务水平和工作条件都得到了明显改善，为教学模式、方法与手段的改革提供了必备的条件。多媒体教室的建设、学生上机时数的增加、实验室建设等一系列措施对教材的建设提出了新的要求。

为了切实体现教育思想和教育观念的转变，依据高等院校教学内容、教学方法和教学手段的现状，机械工业出版社推出了这套“21世纪高等院校应用型规划教材”。

本系列教材以建设“一体化设计、多种媒体有机结合的立体化教材”为宗旨，其目标是：建设一批符合应用型人才培养目标的、适合应用型人才培养模式的系列精品教材。本系列教材的编写者均为相关课程的一线主讲教师，教材内容注重理论与实际应用相结合，其中大力补充新知识、新技术、新工艺、新成果，非常适合各类高等院校、高等职业学校的教学。

为方便老师授课，本套教材为主干课程配备了电子教案、实验指导、习题解答等相关辅助内容。

机械工业出版社

前　言

计算机程序设计语言或计算机程序设计技术课程是高等院校理工类各专业计算机技术基础系列课程中的重要组成部分。计算机程序设计的主要任务就是用合适的计算机程序设计语言对解决问题的方法进行编码处理，即编制程序。C 语言功能丰富、表达能力强、程序执行效率高、可移植性好，既有高级计算机程序设计语言的特点，同时又具有部分汇编语言的特点，因而 C 语言具有较强的系统处理能力。在 C 语言的基础上发展起来的面向对象程序设计语言如C++ 、Java、C# 等都与 C 语言有许多的共同特征，掌握 C 语言对学习和应用这些面向对象的程序设计语言有极大的帮助。

本书作者具有十多年在工程中使用 C 语言进行程序设计的经验，同时具有近 10 年 C 程序设计技术教学的经历。本书从结构化程序设计技术的角度出发，以 C 程序设计语言为载体，通过对 C 语言的基本语法、语义以及学习 C 语言过程中各种常见问题的分析，展现了在计算机应用过程中如何将方法和编码相联系的具体程序设计过程，进而向读者介绍计算机结构化程序设计的基本概念、基本技术和方法。

本书分为 10 章：第 1 章主要讨论 C 语言的基础知识，包括 C 语言的基本组成元素、数据类型、表达式计算、数据类型转换以及 C 程序中的输入输出问题。同时，作为顺序程序设计的典型示例，还介绍了 C 语言数学标准库函数中的典型标准函数的使用方法。第 2 章主要讨论 C 语言中关于分支、循环等各种控制结构语句的使用方法，同时作为应用示例讨论了穷举程序设计方法、迭代程序设计方法以及一元高阶方程的迭代解法。第 3 章主要讨论 C 程序结构方面的基础知识，包括函数的定义、声明和调用；函数的嵌套调用、函数的递归调用以及变量的作用域和生存期等问题。第 4 章主要讨论关于 C 语言中指针的基础知识，包括指针与指针变量的基本概念，指针变量作为函数参数的使用方法，返回指针值的函数，指向函数的指针及其在高阶方程求根通用函数和定积分计算通用函数中的应用。第 5 章主要讨论关于 C 语言数组方面的知识，包括一维数组和二维数组的概念、定义和数组元素的引用方法；数组作为函数参数的使用方法，作为数组应用示例讨论了数组元素值的随机生成、常用的排序和查找方法。第 6 章主要讨论关于字符串处理的基础知识，包括字符串的表示方法；字符串常用处理方法如长度统计、复制、连接、插入和删除等以及常用的字符串处理标准库函数的使用方法。第 7 章主要讨论数组与指针的关系，包括用指针表示数组的方法，指针数组和命令行参数，动态数组的概念、创建和使用方法。第 8 章主要讨论两种特殊的构造数据类型：结构体类型和联合体类型，包括构造数据类型的概念、定义方法和使用方法，结构体类型和联合体类型的异同，单链表的相关处理方法。第 9 章主要讨论位运算和枚举数据类型的基础知识，包括位操作的基本方法和简单应用，位段的概念和引用方法，枚举的概念和简单应用。第 10 章主要讨论关于文件处理的知识，包括文件的打开和关闭，文件内容的读写方法以及简单应用。

本书选用 Visual C++ 6.0 作为教学环境，书中的所有教学示例、习题的参考解答都在 Visual C++ 6.0 集成开发环境中通过测试，为了方便读者学习，在本书的附录中还介绍了用 VC ++6.0 集成环境开发 C 程序的基本方法。

本书每章均提供了与内容紧密相关的习题以方便教学和帮助读者巩固所学知识，本书所有例题、习题解答和电子课件均可以在机械工业出版社网站(www.cmpbook.com)上下载。

限于编者水平，书中错误和不妥之处在所难免，恳请读者不吝赐教。

编　者

目　录

第1章　C程序设计入门

1.1　C语言的发展简史与特点

20世纪70年代初期，随着半导体集成电路技术的发展，计算机硬件系统的性能大大提高，为计算机系统配置的各种软件系统的规模越来越大，软件的结构也越来越复杂。软件的生产组织、管理机制，与极端重视代码运行效率和以程序员个体劳动为基础的传统程序设计方式之间产生了尖锐的矛盾，导致了所谓的“软件危机”的出现。对“软件危机”的解决，引发了对程序设计思想的革命。使传统的重视发挥机器效率的旧程序设计方法逐渐向重视人的因素，强调程序员之间的交流和合作能力，以提高程序的可读性为主要目标的结构化程序设计方法转变。C语言正是在这种时代背景下诞生的。

C语言是国际上广泛流行的一种编译型结构化程序设计语言，它的前身是英国剑桥大学的Martin Richards在20世纪60年代开发的BCPL语言。20世纪70年代初，Ken Thompsom在软件开发中继承和发展了BCPL语言，进而提出了“B语言”并使用它记述和开发了PDP-7小型机上的UNIX操作系统。此后，在美国贝尔研究所进行的更新型的小型机PDP-11的UNIX操作系统的开发工作中，Dennis M.Ritchie和Brian W.Kernighan对B语言进行了进一步的充实和完善，于1972年推出了一种新型的结构化程序设计语言——C语言。

与其他程序设计语言相比，C语言具有下列基本特点：

1）C语言简洁、紧凑，使用方便、灵活。C语言一共只有37个保留字（含C99标准新增的保留字），9种控制语句，区分大小写，程序书写形式自由。

2）C语言是位于汇编语言和高级语言之间的一种程序设计语言。C语言允许直接访问地址，能进行位（bit）运算，能实现汇编语言的大部分功能，可以直接对计算机硬件进行操作。

3）C语言是一种结构化的程序设计语言，程序的逻辑结构由顺序、分支、循环三种基本结构组成。C语言具有结构化控制语句，采用自顶向下、逐步求精的结构化程序设计方法。C语言程序分割为若干个相对独立的功能模块，程序模块之间可以相互调用，并为数据传递提供便利，因此用C语言编制的程序，具有容易理解、便于维护等优点。

4）C语言数据类型丰富。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、联合体（共用体）类型以及枚举类型等。可以实现各种复杂的数据结构，因而C语言具有较强的数据处理能力。

5）C语言运算符丰富。C语言运算符共有34种之多。除一般高级语言使用的+、-、*、/等四则运算及与（&&）、或（||）、非（!）等逻辑运算功能外，还可以实现以二进制位为单位的位与（&）、位或（|）、位非（~）、位异或（^）以及移位（<<，>>）等位运算，并且具有++、--等单目运算和+=、-=、*=、/=等复合运算功能。

6）C语言本身没有提供用于程序中数据输入输出的语句，程序中的数据输入输出通过显示地调用标准库中的输入输出函数实现。

7) C程序开发环境中包含了语言核心、预处理器和标准函数库三个部分，因而在C语言程序中可以使用预处理语句实现宏定义、文件包含以及条件编译等预处理功能。

8) C语言程序可移植性好。C语言本身并不依赖计算机硬件系统，便于硬件结构不同的机型间和各种操作系统间实现程序的移植。

由于C语言具有上述众多特点，近年来迅速地得到广泛普及和应用。适用于各种不同操作系统(UNIX、MS-DOS、视窗系统等)和不同机种的C语言编译系统相继出现，其种类有几十种之多，它们的语句功能基本一致，可以解决C语言程序在不同机种之间的移植问题。但不同版本之间也有某些差异，主要体现在标准函数库中的函数种类、格式和功能上稍有差别。

为了明确地定义与机器无关的C语言，1983年在美国国家标准委员会计算机和信息处理部门之下成立了X3J11技术委员会。1989年通过了所制定的C89(或C90)标准(ANSI/ISO9899:1990，ANSI在1989年公布了标准文档而ISO在1990年重新编排了标准文档章节后才公布)，1999年又制定了新的C语言标准C99。本书以后章节的内容以及C语言语法均参照ANSI C99标准，对只能在支持C99标准的程序设计环境中才能使用的新增语言元素都给与了提示，本书所有示例中选用的库函数也限于标准C下的标准函数库。

1.2 C程序的基本结构

1.2.1 C源程序的组成成分

计算机程序设计语言是人类与计算机进行交流的工具，为了能够在程序员和计算机之间建立一种交流和理解的通道，每一种计算机程序设计语言都有自己特定的语法规则、语义和确定的表现形式，程序的构成规则和程序的书写格式是程序语言表现形式的一个重要方面。下面所讨论的C程序由两个函数main和myputc构成，该程序的功能是：在主函数中从键盘输入一个字符串，依次把串中的每一个字符取出作为参数调用函数myputc，在函数myputc中判断该字符是否为小写字母，若是则将其转换为大写字母，否则不变，然后将其输出到屏幕上，直到字符串中的所有字符处理完为止。为了便于解释，将程序中的每一行都加上了行号。

【例1-1】C语言源程序的组成成分和基本结构。

```
1  /* Name: e0101.cpp
2     Function: Print string as uppercase */
3  #include <stdio.h>
4  #define SIZE 80
5  void main()
6  {
7      void myputc(char ch);   /* 函数myputc的原型声明 */
8      char str[SIZE];
9      int j;
10     gets(str);      /* 从键盘上接收一个输入字符串 */
11     for(j=0;str[j]!='\0';j++)/* 依次取出串中的字符作为参数调用函数myputc */
12         myputc(str[j]);
13 }
14 /* 函数myputc的定义 */
```

```
15  void myputc(char ch)
16  {
17      char cc;
18      cc=(ch>='a'&&ch<='z')? ch+'A'-'a':ch;
19      putchar(cc);
20  }
```

C程序是函数型程序结构。根据程序所解决的问题不同,一个C程序可以由一个函数构成,此时这个函数的名字只能是main。C程序也可以由一个主函数和若干个其他函数组成,除主函数的名字必须为main之外,其他函数由程序员根据函数所实现的逻辑功能予以命名,如例1-1中的myputc。C程序中,必须包含一个且只能有一个名字为main的主函数,C程序的执行是从主函数开始的,主函数中的所有语句执行完毕,则程序执行结束。

C程序中,一个函数一般实现一个相对独立的逻辑功能。函数由函数首部(函数头)和函数体组成,例如本例中主函数main由第5~13行组成,其主要工作是从键盘上接收输入的字符串数据,然后调用函数myputc对其进行相应的处理,函数中第5行是函数首部,第6~13行是其函数体;函数myputc由第15~20行组成,其主要工作是判断从主函数中传递过来的字符数据是否为小写字母,若是则将其转换成大写字母输出,否则按原数据输出,其中第15行是函数首部,第16~20行是其函数体。关于C语言中函数的其他问题,将在第3章中讨论。

注释语句的主要功能是对程序做一些注解性的工作,注释语句对程序的功能没有任何影响。C语言中注释语句的构成方式为:

```
/*<字符序列>*/
```

注释语句中的字符序列可以由一行字符组成,如例1-1中的第14行。字符序列也可以由若干行字符组成,如第1~2行;注释语句可以出现在程序中的任何地方,如第7、10、11行中的注释语句。但需要特别注意的是,注释语句不能插入到在语法上是一个基本整体的程序构成元素中,例如下面的注释方法是错误的:

```
g/*从键盘上接收一个输入字符串*/ets(str);
```

其原因是gets在程序的语法结构上是一个整体,不允许将其分开。

注释语句在程序设计中的另外一个重要作用体现在程序的调试过程中,如在调试程序时认为程序中的某些部分是不需要的或者是错误的,较好的解决方法不是直接将这些部分去掉,而是在这些部分的前后分别加上注释语句的标记,使其在程序中作为注释出现而不起功能性的作用。这样当发现这些部分在功能上有需要时,只需去掉相应的注释符号即可恢复其功能。例如,若认为例1-1中的第4行是不需要的,则可以通过下面的形式使其失去功能,而作为注释语句出现在程序中:

```
/*  #define SIZE 80  */
```

在C99标准中,提供了另外一种书写注释语句的格式:"//<字符序列>"。例如对于例1-1中的第10行可以写为:"gets(str);//从键盘上接收一个输入字符串",第14行可以写为:"//函数myputc的定义"。需要注意的是使用"/*<字符序列>*/"方式,可以将注释写在若干行上,而使用"//<字符序列>"方式则只能将注释写在一行上,两种注释方法在程序中可以

根据需要混用。另外，由于“//<字符序列>”方式也是C++语言支持的注释方式，所以即使在不支持C99标准的C/C++编译环境中，也可以使用这种注释方式。

预处理语句是程序在编译之前就被执行的语句，其语句特征是用“#”字符开始，如例1-1中的第3行和第4行。第3行处理语句的意思是将文件stdio.h嵌入到该语句处，以实现对程序中标准输入库函数gets的声明；第4行的意思是定义在程序中用单词SIZE表示数据80，即程序中该语句之下的所有SIZE在编译时均被代换为80进行处理。关于编译预处理的其他问题，将在第3章中予以讨论。

C语言是一种强制定义(声明)的程序设计语言，任何在程序中处理的数据对象和要调用的函数都需要在使用之前预先定义和声明。定义或声明的形式与处理的对象有关，如例1-1中的第3行通过预处理语句对标准库函数gets进行了声明；第7行对自定义函数myputc进行了声明；第8行定义了一个字符数组str；第9行定义了一个整型数据对象(变量)j。

C语句用分号“;”作为其结束符号，每一个完整的C语句都需要用分号结尾。但需要注意的是预处理语句并不是C语句，所以预处理语句不需要使用分号结尾。

1.2.2 C语言的基本元素

1. C语言的字符集

每种计算机程序设计语言都规定了在书写源程序时允许使用的字符集，以便语言处理系统能正确识别它们。C语言规定书写C源程序的字符集由以下字符组成：

- 小写英文字母：a,b,c,……,z
- 大写英文字母：A,B,C,……,Z
- 数字：0,1,2,3,……,9
- 特殊字符：+ = - _() * & % $! | < > . , ; : " ' / ? { } ~ [ ] ^
- 不可印出字符：空格，换行，制表符等

2. 标识符

标识符是程序中处理的数据对象(如变量、常量、函数、数据类型等)的名字。标识符的命名规则是：

1) 组成标识符的字符为字母、数字和下划线。

2) 标识符中第一个字符必须是字母或下划线。

3) 多数C编译系统在构成标识符时都要区分字母的大小写，即abc和Abc是不相同的标识符。

4) 构成标识符的字符个数(标识符长度)与所使用的环境相关，但C89规定可以区分的最大长度为31个字符，C99规定的可以区分的最大长度是63个字符。例如在C89中，abcdefghijklmnopqrstuvwxyzabcdefg和abcdefghijklmnopqrstuvwxyzabcdef被认为是相同的标识符，而在支持C99标准的环境中它们则是不同的标识符。

标识符分为两大类：系统保留字和用户标识符。C语言有37个系统保留字(又称为关键字)，保留字是一类特殊的标识符，是C语言中具有特定严格意义的基本词汇，任何情况下都不能将它们作为用户标识符使用。下面列出C语言中的保留字(其中标有星号上标的是在C99标准中增加的保留字)：auto、_Bool*、break、case、char、_Complex*、const、continue、default、do、double、else、enum、extern、float、for、goto、if、_Imaginary*、inline*、int、long、register、re-

strict*、return、short、signed、sizeof、static、struct、switch、typedef、union、unsigned、void、volatile、while。

下面几个标识符从严格意义上说不属于系统保留字，它们常出现在C的预处理器中，C语言处理系统中为它们赋予了特定的含义，建议用户不要将它们在程序中随意使用，以免造成混淆，这些标识符是：define、undef、include、ifdef、ifndef、endif、line、error、elif、pragma。

程序员(用户)在程序中自定义标识符时，除了必须遵守标识符的命名规则外，还需要注意以下两个方面：一是要使标识符的名字既有意义，又便于阅读；二是要注意避免含义上或书写上引起混淆。

下面是一些合法用户自定义标识符的例子：a、b1、file_name、_buf。

下面是不合法的用户自定义标识符的例子及错误原因：

```
123abc          /* 不是以英文字母开头 */
float           /* 与系统保留字同名 */
up.to           /* 标识符中出现了非法字符"." */
zhang san       /* 标识符中间出现了非法字符空格 */
```

3. 函数

在C程序中，函数是构成程序的基本模块，每个函数具有相对独立的功能。C程序中使用的函数有三种：主函数(即main()函数)、C语言编译系统提供的标准库函数和用户自定义的函数。主函数是C程序执行的入口，即程序总是从主函数中的第一个可以执行的语句开始执行；一般情况下也是程序执行的出口，即在执行完了主函数中的所有语句后程序结束。标准库函数是语言处理系统提供的常用功能的处理程序代码，标准库函数经过了严格的测试和优化，合理地选用标准库函数可达到事半功倍的程序设计效果。在标准库中，标准库函数的原型声明被分门别类地书写在对应的头文件中，所以在程序中若要使用标准库函数，则需要在程序中合适的地方(调用标准库函数之前)用文件包含预处理语句将与所使用库函数相应的头文件包含到程序中来，如例1-1中的第3行"#include <stdio.h>"。用户自定义函数即程序员根据所设计应用程序的功能自己编写的函数，合理地编写用户自定义函数，可以简化程序模块的结构，便于程序的阅读和调试，是结构化程序设计方法的主要内容之一。关于自定义函数的相关问题，将在第3章中予以讨论。

4. C程序书写的基本要点

1) C程序习惯上使用小写英文字母。为了清晰起见，在C程序中往往使用大写英文字母来表示宏定义或其他具有特殊意义的标识符。

2) C程序中不强调程序行的概念。一行中可以有多条语句，一个语句也可以写在多行上，但语句与语句之间要用分号";"分隔。

3) C程序为了增强程序的可读性，可以使用适量的空格、空行和适当的行间缩进结构。但要注意，程序中的变量名、函数名以及C语言本身使用的单词(如保留字、语句结构等)，不能在其中插入空格。

1.3 C语言的基本数据类型

1.3.1 C语言数据类型概述

数据是程序的必要组成部分，也是程序的处理对象。C语言提供的数据类型比一般高级语言丰富，除整型、实型、字符型等基本类型外，C语言还提供了数组、指针、结构体、联合体、枚举、位、位段等数据类型，可以使用这些数据类型构成复杂的数据结构。C语言提供的数据结构以数据类型的形式出现，C语言的数据类型如图1-1所示，其中带星号的数据类型为C99新增数据类型。

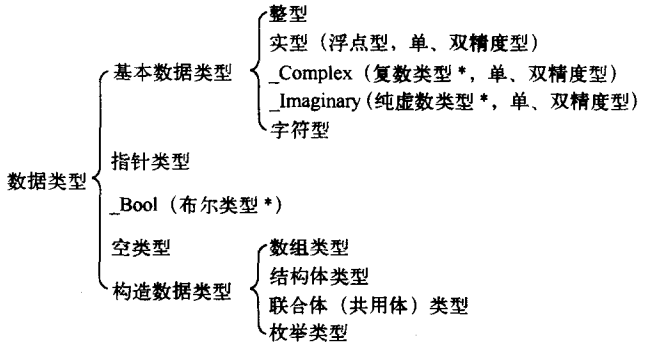


图1-1 数据类型

C程序中处理的数据有常量与变量之分，它们均分别属于上述类型。

在程序的运行过程中，其值不能被改变的量称为常量。C语言的常量有三类：数、字符和字符串，它们不必进行任何说明就可以在程序中直接使用。在C程序设计中，还可以使用标识符来代表一个常量，称为符号常量。与变量不同，符号常量的值在其作用范围内不能被改变，也不能再被赋值，如例1-1中的标识符SIZE。符号常量一般习惯用大写字母表示，以便于与程序中的变量区别。

在程序的运行过程中，其值能够被改变的量称为变量，变量用标识符来表示。C语言中规定，程序中的变量在使用之前必须加以定义，对变量的定义可以出现在函数前面或内部，也可以在函数的参数说明部分或复合语句的说明部分。程序中的每一个变量都应有确定的数据类型，在一个程序中一个变量只能属于一个类型，不能先后被定义为两个或多个不同类型。其原因如下：

1）一种数据类型对应着相应的取值范围。

2）一种数据类型对应着一组允许的操作。

3）不同数据类型的数据在计算机内存中采用不同的表示方法，占据不同长度的存储区。

变量定义的一般形式如下：

<数据类型名> <变量名表>;

其中数据类型名用于指明变量名表中变量所具有的数据类型，变量名表由一个变量或若