

C语言与 程序设计

夏宽理 编著



复旦大学出版社

C LANGUAGE & PROGRAMMING

C 语言与程序设计

夏宽理 编著

复旦大学出版社

(沪)新登字 202 号

责任编辑 林溪波
责任校对 陆宏光

C 语言与程序设计

夏宽理 编著

复旦大学出版社出版

(上海国权路 579 号)

新华书店上海发行所发行 上海永南印刷厂印刷

开本 787×1092 1/16 印张 19.25 插页 0 字数 500,000

1994 年 5 月第 1 版 1994 年 8 月第 2 次印刷

印数 5001—15,000

ISBN7-309-01346-8/T·108

定价:15.00 元

内 容 简 介

本书详细介绍了C语言的基本概念和用C语言进行结构化程序设计的方法和技巧。

本书的特点是叙述深入浅出、体系合理、概念准确、清晰,注重程序设计能力的训练,采用逐步求精的程序开发技术介绍程序开发过程。

本书可作为高等院校程序设计课程的教材和教学参考书,或作为应用软件水平考试及计算机培训班的教材或参考书,也可作为计算机有关专业的程序设计课程的自学用书。

JS A2 / 15

前 言

C语言是目前国内外得到广泛使用的一种程序设计语言,C语言具有功能丰富、表达能力强、使用灵活方便、便于大型程序开发和编写可移植性好的程序等优点。由于C语言引入了反映计算机硬件特性的机制,使C语言也能编写直接控制计算机硬件设备的程序。所以C语言是一个非常适合于写系统软件的语言。随着计算机应用的日益深化,大量的应用软件也常用C语言编写,C语言已被广大计算机应用人员所使用。

由于C语言涉及的概念比较丰富,以及它描述的多样性和使用的灵活性,会给初学者完整掌握C语言带来一定的困难。本书力求在有关概念介绍时叙述准确、循序渐进,有助于读者准确、深入理解概念,全面掌握C语言。

对广大读者来说,学习C语言的目的是用它来编写程序。为能熟练地进行程序设计,除需掌握一种程序语言之外,还应有算法、数据结构和程序设计方法等方面的知识。而算法、数据结构等方面知识的学习又基于程序语言和基本的程序设计方法和技巧。本书作为计算机程序设计课程的教材,在详细介绍C语言同时,特别注重于C语言的使用、正确运用C语言编写程序和程序算法设计及程序设计过程的介绍。使读者通过对本书的学习,不仅正确了解C语言,还能掌握初步的程序设计方法和技巧,在程序设计能力方面受到良好的训练。

本书第十一章的程序设计实例是为了介绍程序的开发过程而编写的。当用本书作为程序设计课程的教材时,限于授课时间,可只讲授其中一、二个实例,其余让读者自习。第十二章的TURBO C 屏幕作图方法简介是为了让读者了解计算机作图程序的编制方法而编写的,其中内容虽开辟了一个新的应用领域,然而都是由一些基本函数实现的,并有规范的编程方法,读者只需模仿实例程序,通过自学掌握有关内容。

对于C语言,本书主要介绍其最基本最常用的内容,由于C语言的有些内容涉及到非常深入的细节,且不同版本在实现上也有所差异,本书未对那些内容作深入全面的讨论。读者在学习时,不要过于拘泥于每个细节。书中所举例子基本上是大多数C语言版本所共有的,如有的系统在使用某些例子时,有细小差别,请读者自行修改。

本书主要是为大学计算机系的学生学习计算机程序设计课程而编写的,讲授时间约为72学时,每位学生另有30小时以上的上机时间。当用本书作为计算机水平考试补习班或各种计算机培训班,包括自学辅导班的教材时,那些在目录中标有星号的章节,因涉及C语言和程序设计能力方面更深入的内容,可适量少讲或不讲。

由于作者水平有限,文中欠妥及谬误之处,敬请读者指正。

在本书编写过程中,得到多位老师的关心和支持,特别是徐君毅和李宗葛两位老师,对本书的编写提出了许多宝贵意见,并详细审阅了本书,在此深表谢意。

作 者

1993年7月于复旦

目 录

第一章 C语言概述	1
§ 1.1 C语言的历史背景和特点	1
§ 1.2 C语言初步知识	2
§ 1.3 C程序的编译和运行	8
习 题	10
第二章 基本数据类型、运算符和表达式	11
§ 2.1 基本数据类型.....	11
§ 2.2 运算符和表达式.....	18
习 题	30
第三章 数据输入和输出	31
§ 3.1 字符输出函数.....	31
§ 3.2 字符输入函数.....	32
§ 3.3 格式输出函数.....	33
§ 3.4 格式输入函数.....	36
习 题	39
第四章 语句	41
§ 4.1 基本语句.....	41
§ 4.2 结构化控制结构.....	43
§ 4.3 控制结构语句.....	46
习 题	58
第五章 数组	62
§ 5.1 数组的基本概念.....	62
§ 5.2 一维数组.....	63
§ 5.3 多维数组.....	69
§ 5.4 字符数组和字符串.....	74
§ 5.5 数组程序实例.....	79
习 题	87
第六章 函数与程序结构	91
§ 6.1 函数定义和函数调用.....	91
§ 6.2 递归函数	100
§ 6.3 程序实例	106
§ 6.4 作用域与存储类	118

习 题.....	128
第七章 编译预处理	130
§ 7.1 宏定义	130
§ 7.2 文件包含	135
§ 7.3* 条件编译	136
§ 7.4* 行控制	139
习 题.....	140
第八章 指针	141
§ 8.1 指针和地址	141
§ 8.2 指针和数组	148
§ 8.3 返回指针值的函数	156
§ 8.4 指针数组和指向指针的指针	158
§ 8.5* 指向函数的指针	165
习 题.....	171
第九章 结构与联合	173
§ 9.1 结构	173
§ 9.2 结构数组与指向结构的指针	178
§ 9.3 动态数据结构	185
§ 9.4 联合	202
§ 9.5* 位域	204
§ 9.6* 枚举类型	206
§ 9.7 类型定义	209
习 题.....	211
第十章 文件	213
§ 10.1 文件概述.....	213
§ 10.2 文件的使用.....	214
§ 10.3 程序实例.....	224
习 题.....	237
第十一章* 程序设计实例	238
§ 11.1 堆积排序.....	239
§ 11.2 找路径.....	243
§ 11.3 搜索找解.....	248
§ 11.4 表达式解释计算.....	251
习 题.....	256
第十二章* TURBO C 屏幕作图方法简介	258
§ 12.1 屏幕图形显示基础.....	258
§ 12.2 基本图形绘制函数.....	261
§ 12.3 图形填充函数.....	264
§ 12.4 视口和调色板.....	265

§ 12.5	图形复制和保存	267
§ 12.6	正文输出	268
§ 12.7	字符屏幕管理	269
§ 12.8	作图函数应用实例	271
习 题		283
附录 1	常用字符与 ASCII 代码对照表	284
附录 2	运算符和结合性	285
附录 3	C 语言常用语法提要	286
附录 4	C 系统常用库函数	290
参考资料		298

第一章 C 语言概述

§ 1.1 C 语言的历史背景和特点

一、C 语言的历史背景

C 语言是被广泛应用的计算机高级程序设计语言。能用它编写各种复杂的应用程序，也能用它编写包括操作系统在内的系统程序。

因操作系统等系统程序依赖于计算机硬件，以前这类系统程序主要用汇编语言编写，因而程序的可读性和可移植性都很差，严重妨碍了系统程序的生产。为此，人们努力寻求一种程序语言，使它既具有高级语言的特性，能够编写可读性高、便于移植的程序，改善编写程序的环境；又具有某些必要的汇编语言特性，能描述对硬件的操作。例如能对内存地址的操作，对位的操作，对字的移位操作，及对寄存器操作等。C 语言就是在人们寻找集高级语言和汇编语言优点于一身的高级语言过程中产生的。

C 语言是在 B 语言的基础上发展起来的。1963 年，英国剑桥大学在 ALGOL₆₀ 程序设计语言基础上，推出了 CPL (Combined Programming Language)。因其规模大，实现难等原因，1967 年 Martin Richards 对 CPL 作了简化，推出了 BCPL (B 代表 basic)。以后，1970 年美国贝尔实验室的 Ken Thompson 又在 BCPL 的基础上，再次作进一步的简化，设计出简单又接近硬件的 B 语言，并用 B 语言写了 UNIX 操作系统和大量的实用程序。因 B 语言只有单一的字类型，过于简单等原因而未能流行。D. M. Ritchie 从 1971 年开始在 B 语言基础上设计了 C 语言，于 1972 年投入使用。1973 年 K. Thompson 和 D. M. Ritchie 把 UNIX 系统用 C 重写了一遍，增加了多道程序设计功能，使整个系统，包括 C 语言的编译程序都建立在 C 语言的基础上。第五版 UNIX 系统 (UNIXV₅) 奠定了 UNIX 系统的基础。到 1975 年，UNIX 第六版问世。随着 UNIX 的巨大成功和被广泛移植到各种机器上，C 语言也被人们所接受，并移植到大、小、微型机上。C 语言已风靡全世界，成为世界上应用最广的计算机程序设计语言之一。

以 1978 年发表的 UNIX 第七版中的 C 编译程序为基础，Brian W. Kernighan 与 Dennis M. Ritchie 合著了《The C Programming Language》一书。这本书介绍的 C 语言成为以后 C 语言版本的基础。1983 年，美国国家标准化协会 (ANSI) 对 C 语言的各种版本作了扩充和完善，制定了 C 的标准，称为 ANSI C。本书的叙述基本上以 ANSI C 为基础。目前广泛流行的各种版本 C 语言编译系统在非主要部分中稍有不同。在微型机上使用的主要有 Microsoft C、TURBO C、Quick C 等。因不同版本各有差异，因此读者在参考本书例题编写习题上机时应参阅有关手册，了解你使用的计算机系统的 C 编译系统的特别规定。

二、C 语言的特点

因 C 语言集中了一般高级语言的优点和汇编语言的优点，能用它方便地编写不依赖于计算机硬件设施的各种应用程序，又能用它编写包括操作系统在内的各种系统程序。C 语言

具有多方面的特点，其主要特点有以下几个方面。

1. 语言表达能力强

C语言包含丰富的运算符，有的运算符反映了当前计算机的性能，包含可直接由硬件实现的算术逻辑运算，有效到足以取代汇编语言编写各种系统程序和应用程序。众多的运算符使C的运算类型极其丰富，可以表达数值运算、字运算、位运算、和地址运算等。

2. 具有数据类型构造能力和结构化的程序控制结构

C语言能在字符、整数、浮点数等基本类型基础上按结构化的层次构造方法构造数组、结构和联合等各种结构化的数据类型。特别是C的指针类型的灵活多样，非常有助于构造链表、树、栈、图等复杂的数据结构。另一方面，它的结构化程序控制结构符合结构化程序设计的要求，可编写结构非常好的程序。此外，它的数据的静态和外部存贮类机制有助于信息隐蔽、抽象的模块化结构程序设计。

3. 语言简洁、紧凑、使用方便灵活

用C语言编写的程序通常比用其他高级语言编写的程序更简练，代码行少。语言的许多成分都通过显式函数调用完成。C语言没有I/O设施，也没有并行操作、同步或协同程序等复杂控制。另外，C语言程序在运行时所需要的支持少，占用的存贮空间也小。

4. 能使编译程序产生执行效率高的代码

一个高级语言能否用来描述系统程序，除语言表达能力之外，还有能否产生高质量的代码这个重要因素。许多高级语言相对汇编语言而言其代码的执行效率要低得多。但C语言则不然，许多试验表明，用C语言描述较汇编语言描述，其代码执行效率只低约10—20%，而用C语言编程比汇编语言编程迅速得多，程序的可读性又高，特别是C语言程序比较容易移植。所以C语言成了人们描述系统程序和应用程序比较理想的工具。

5. 用C语言可编写移植性较好的程序

程序的可移植性是指在一个环境上运行的程序可以不加或稍加改动后在另一个完全不同的环境上运行。汇编语言是依赖于机器硬件的，用汇编语言编写的程序不可移植。而有些高级语言，因它们的编译程序不可移植，影响了用它们编写的程序的可移植性。目前在许多机器上都有C编译系统，且大部分是由C语言编译移植得到的。由于C语言的编译程序便于移植，也就提高了C程序的可移植性。

C语言有诸多优点，但也有一些不足之处。用C语言编程，自由度大，如对变量的类型约束不够严格，整型、字符型及逻辑型数据的通用，指针和数组的通用等。过多的通用性限制了编译程序对C程序作充分的句法和语义检查，可能会无视某些使用上的失误，依旧能正常编译，不能及时发现程序中的错误，给程序的调试和排错造成一些困难。另外，C语言的运算符优先级太多，不便于记忆，有些还与常规约定有所不同；类型检验弱，数据类型转换比较随便等，影响了程序的安全性。

§ 1.2 C语言初步知识

本节从C程序例子出发介绍C语言的初步知识，首先让读者对C语言有一个初步的印象。其中提及的概念、名称在以后的章节中都将有详细的介绍。对于已有其他高级语言使用经验的读者，只要浏览阅读一遍就行，可能有些概念的提法与你熟悉的语言稍有不同。

一、C 程序结构

现用简单的 C 程序说明 C 程序的结构。

【例 1.1】

```
/* ex-1-1.c 一个只输出一行信息的 C 程序 */
main() /* 主函数 */
{
    printf(" This book is <Programming with C language>. \n");
}
```

该程序只输出一行信息：

This book is <Programming with C language>.

一个 C 程序一般包含有一个或多个函数，在程序的所有函数中必须有一个且只有一个名为 main 的函数，称它为主函数。如例 1.1 程序只有一个主函数。组成程序的多个函数可存放在一个或多个源程序文件中。例 1.1 的程序假定存放在 ex-1-1.c 文件中（习惯将含有函数定义的程序文件名称的后缀定为 c）。在函数名之后有一对圆括号“()”，圆括号可包含函数的参数，参数数目可根据需要而定，可有可无。函数体用花括号“{}”括起来。花括号可以用来括起任何一组 C 代码，从而构成复合语句或分程序。例 1.1 的函数体只有一个 printf() 函数调用。printf() 函数是 C 系统的函数库中的输出函数（详见第三章），其中用双引号“”括住的字符串指明 printf() 函数的输出格式。例中的输出格式表示以字符串原样输出，而“\n”是换行的意思，即输出字符串“This book is<Programming with C language>.”之后换行。简单 C 语句之后有一个分号“;”，它是某些语句的结束符。程序中的“/* ... */”表示程序的注释部分。程序中的注释是给阅读程序的人看的，对程序编译和运行都没有作用。为便于理解，本书中的注释大量使用中文。注释写在程序的开头，以说明该程序的文件名称、程序的功能、使用方法、最后更改日期等；注释出现在程序代码行中或某行语句之后，用来说明一段程序代码或一个语句的功能、意义等。

【例 1.2】

```
/* ex-1-2.c 读入两个整数,输出它们的和 */
main()
{
    int x, y, sum; /* 定义变量 x, y, sum */
    printf(" Input x and y\n"); /* 提示输入数据 */
    scanf("%d%d", &x, &y); /* 输入 x 和 y 的值 */
    sum = x+y; /* 求 sum = x+y */
    printf(" x+y = %d\n", sum); /* 输出结果 */
}
```

例 1.2 的作用是输入两个整数，求它们的和并输出。例中的第四行是变量定义部分，定义三个整型变量，分别命名为 x, y 和 sum。第五行实现输出“Input x and y”字样，提示用户程序开始执行，请输入 x 和 y 的值。第六行的 scanf() 是 C 函数库中的输入函数（详

见第三章)。其中“%d”是十进制整数输入格式，用来指定输入数据的数据类型和格式。这里表示将用户键入的数据以十进制整数理解，将它转换成整型量的机内表示。&x 和 &y 中的“&”的含义是取地址。对于本例，意指顺序输入的两个十进制整数将它们分别存于用 x 和 y 命名的变量所对应的内存单元中。直观地理解就是输入值给变量 x 和 y。第七行是完成 x+y 的计算，并将结果赋给变量 sum。第八行的 printf () 函数调用将输出“x+y =”字样和 sum 的值，并换行。其中十进制整数输出格式“%d”意指将 sum 的整数值按十进制整数形式输出。该程序运行情况如下：

```
Input x and y
12 15      (假定输入 12 和 15 两个数)
x+y = 27
```

其中第 1、3 行是程序输出的，第 2 行是用户键入的，12 与 15 之间用空格分隔。

【例 1.3】

```
/* ex-1-3.c 输入两个实数，输出它们中的小的数 */
main()
{
    float x,y,c;      /* 变量定义 */
    float min();     /* 函数说明 */
    printf("Input x and y. \n");
    scanf("%f%f",&x,&y);
    c = min(x,y);    /* 调用函数 min() */
    printf("MIN(%f,%f) = %f\n",x,y,c);
}
/* 以下定义函数 min() */
float min(a, b)
float a, b;        /* 函数形参 a, b 的说明 */
{
    float temp;    /* 函数使用的变量的定义 */
    if (a < b)
        temp = a;
    else
        temp = b;
    return temp;  /* 返回 temp 到调用 min () 函数处 */
}
```

本例子程序包含两个函数：主函数 main () 和求两个实数中小的数的函数 min ()。函数 min () 的作用是将 a, b 中的较小的值赋给变量 temp，然后用返回语句 (return) 将 temp 的值返回给对它的调用处 (主函数中的 c = min (x, y))。本例 main () 函数中的第八行为调用 min () 函数，在调用时将实参 x 和 y 的值分别传送给 min () 函数中的形参 a 和 b (实参的值传送给形参是函数调用机制自动完成的，详见第六章)。经执行 min () 函数后，得到一个值，这个值被赋给变量 c。程序中的输入输出函数调用中的格式“%f”与前面例子

中介绍的格式“%d”相似，格式“%f”是用于输入输出浮点型数据的。请注意主函数中的函数说明“float min ();”；它是用于说明名字 min 是一个返回浮点型(float)值的函数名。C语言中遵守对象先说明或定义，后使用的原则。若主函数未对函数 min () 作说明，则主函数第八行中的 min () 函数被省缺设定为整型函数，而后面的 min () 函数定义却是浮点型，导致类型不一致的错误。另请读者注意C语言中说明与定义的区别，直观的理解是说明只给一个名赋予某种意义，如变量说明只宣布变量的类型等特性、函数说明只宣布它的结果类型等特性；而定义除表明全部特性外，更主要的是为对象确定了实体，如变量定义要求为变量分配存贮，函数定义更主要的是指定参数和实现其功能的程序代码。

通过以上例子，对C程序的结构至少有以下几方面的初步认识。

(1) C程序的执行代码由函数构成。C程序至少包含有主函数(main () 函数)，还可包含其他函数。C系统提供丰富的库函数。另外，一个C程序可由多个源程序文件组成，每个源程序文件可有一个或多个函数组成，C语言的这个特点有助于程序的模块化设计。

(2) 一个函数定义由函数头和函数体组成。函数头包括函数属性、函数类型、函数名、函数形式参数名，形式参数类型。

一个函数可以没有参数，但函数名之后的一对圆括号是必需的。

函数体是函数头之后用一对花括号括住的部分。函数体用于描述实现函数功能的代码，它一般又可包括：

- 说明和定义部分。说明数据结构(类型)和定义函数专用的变量等。
- 执行部分。由C语句和控制结构代码组成，是详细描述实现函数功能的C代码。

在某些情况下，函数体没有说明和定义部分。甚至可以没有执行部分。没有任何内容的函数体也是合法的。允许这种函数的合法性，便于大程序的一边开发，一边调试，逐步开发完全程序。

(3) C程序从 main () 函数开始执行，不管 main () 函数在程序中的位置如何。在大型C程序设计中，通常 main () 函数独占一个文件。并在该文件中用注释详细说明该程序的功能和使用方法。

(4) C程序的书写格式是自由的，即一行可写多个语句，一个语句也可分写在多行上。为了便于人们(包括自己)阅读你的程序，建议用一种良好的风格书写你的程序。本书采用的程序书写风格就是一种较好的书写风格。

(5) C语言的每个简单语句，说明及变量定义之后都必须以分号结尾，分号是它们必要的组成部分。在本书给出的C语言句法成分的一般形式中，若最后有分号，则该分号是它的必要组成部分。

(6) C语言本身未提供输入输出语句，但通常提供的C系统包含有函数库。函数库十分丰富，包括功能齐全的输入输出函数(如 printf () 函数和 scanf () 函数)。标准C系统提供一百多个库函数，而 TURBO C、MS C 4. 0 提供了三百多个库函数。

(7) 在C程序的任何部分都可插入注释。注释便于人们理解和阅读C程序，但有一点需要特别指出，C语言中，注释不允许嵌套。在一个注释内，不能再有“/* ... */”。

二、C语言基本概念

1. 基本符号和标识符

任何一种高级语言，都有自己的基本词汇表。C语言的基本词汇表有下列几部分：

- 数字 10 个 (0~9)。
- 英文字母大、小写各 26 个 (A~Z, a~z)。
- 下划线字符 “_”，这个下划线字符起一个英文字母的作用，以构成标识符等语法成分。
- 特殊符号。主要包括运算符和关键字。其中运算符见附录 2。C 语言为了清晰表达程序成分的意义，使用了一些英文单词，这些单词称为关键字。有：

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

下面几个字虽不属于关键字，但建议读者把它们看作关键字，不要在程序中随便使用。这些字是

define, undef, include, ifdef, ifndef,
endif, line.

这些字用在 C 语言的预处理程序中。

· 标识符。为标识变量、常量、类型、函数、语句等程序成分对象，需要为它们命名，C 语言用标识符给它们命名。在 C 语言中，一个合理的标识符由英文字母或下划线符开头，后跟或不跟由字母、下划线符、数字组成的字符列。一般以下划线开头的标识符作内部使用。标识符作为程序成分对象的名称，建议读者给程序成分对象命名时，使用能反映该对象意义的标识符，便于联想和记忆。另外，请注意不同的 C 系统对标识符的有效字符个数有不同的规定。一般取 8 个字符，也就是说，两个超过 8 个字符的不同标识符，当前 8 个字符完全一样时，系统就认为它们是同一个标识符，而不加以区别。

利用基本词汇表的符号和关键字，按照给定的 C 语言的句法规则就可进一步命名对象，构造语句、函数，直至整个程序。

2. 数据类型

一个计算机程序主要包括两方面的内容：一是关于程序实现的算法的操作步骤描述，即动作描述；二是关于算法操作对象的描述，即数据描述。程序设计就是考虑和设计数据结构和算法，正如著名计算机科学家 N. Wirth 所提出的“数据结构+算法 = 程序”。程序设计的基本方法是抽象、枚举和归纳。抽象包括算法的抽象和数据抽象。算法抽象是指算法的寻求（或开发）采用逐步求精、逐层分解的方法。数据抽象也指在算法抽象的过程中还逐步完善数据结构和引入新的数据及确定关于数据的操作。

在高级语言中，引入数据类型的概念，数据类型反映两方面的内容：它的数据能被如何表示（它的所有表示构成了数据类型的值的集合）和如何处理它的值（它的所有处理方式构成了数据类型的操作集合）。例如整型是系统预先设定的有限整数集和一个关于整数的运算集。高级语言为便于编写程序，预先设定了若干基本数据类型；又为了使程序能描述处理现实世界中各种复杂数据结构的问题，提供若干从基本数据类型出发构造各种复杂数据结构的手段，且这种手段能被反复应用。

C语言包含的基本数据类型有：

整型 (short, int, long), 实型 (float, double, long double), 字符型 (char)。

能构造复杂数据类型的构造设施有：

数组、结构、联合。

使用以上数据类型构造设施，从基本数据类型和指针类型出发，构造出各种复杂的数据类型。以上构造设施能被反复应用，习惯以最终使用的构造设施称呼得到的数据类型。如数组类型、结构类型、联合类型。另外，还有枚举构造设施用于说明一组有关的符号（标识符）为某类型的值的方法。

为描述数据实体之间具有某种关系的数据结构，如链表、树、图等数据结构，C语言引入了指针类型。在C语言中，指针类型直接赋予数据对象在内存中的地址的概念，使指针的使用非常灵活、多样。

3. 常量

在程序运行过程中，其值不能改变或不允许改变的数据对象称为常量。常量按其值的类型区分它的类型，如 15、0、-7 是整型常量；5.0、-12.36 为浮点型常量；而 'a'、'b' 为字符型常量。

在程序中直接使用字面形式的常量，在某些情况下会有不便之处。如表示同一意义的常量在程序中出现多处，会不易保持多处出现的一致性，也不便修改。如程序中多处使用 π 的值，不同之处书写的位数可能不易保持相同。当为了提高计算精度时，就需要多处作同样修改，容易产生遗漏。另外，字面形式的常量出现在程序中，不能反映该值的意义。给常量命名是解决上述问题的好方法。程序用常量的名引用常量，保证一致性，也便于修改。如为常量命名一个反映常量意义的名，能提高程序的可读性。用宏定义给常量命名，其一般形式之一是

```
#define 标识符 字符列
```

如

```
#define PI 3.14159
```

```
#define MAXN 100
```

#define 是C语言的预处理命令，详细用法见第七章。注意 #define 命令之后无需另用分号结束。另外，程序中不允许对该标识符赋值。

4. 变量

在程序运行过程中，其值可以改变的数据对象称为变量。变量在它存在期间，在内存中占据一定的存储单元，以存放变量的值。可以给变量一个名字，但请注意变量名、变量所占据的内存地址、变量的值等概念的区别。

程序在使用变量之前，先要对变量作定义。如例 1.2 和例 1.3 所示。变量定义指定变量的名字和数据类型，其中名字以便程序对它引用；数据类型给编译系统以信息，为它分配相应的存储单元，检查对变量操作的合理性，并为合理的操作生成正确的目标代码。

变量定义的一般形式是

类型 变量名表；

其中变量名表由一个或多个变量名组成。例如

```
int i, j, sum ; /* 定义三个变量 */
```

```
char str [100] ; /* 定义一个字符数组 */  
float z ; /* 定义一个变量 */
```

在 C 语言中，变量定义在为变量指定类型和名称的同时，还可以为变量指定初值。为变量指定初值称作变量初始化。例如：

```
int index = 100;
```

有关变量初始化的方法、限制等细节见有关章节。

§ 1.3 C 程序的编译和运行

本节对于初次上机的读者是有帮助的。在编写一个 C 程序后，要上机运行，大致经历编辑、编译、连接和执行四个阶段。不同的 C 系统环境，使用的方式各有不同，有不同的实用程序辅助用户解决上机过程中的复杂问题。这里简要介绍两种不同环境下的 C 程序编译和运行的方法，详细内容请读者参考有关环境的使用说明书。

一、在 UNIX 操作系统下编译运行 C 程序

1. 编辑源程序文件

为源程序文件命名一个名字（设为 example. c）。利用屏幕编辑程序 vi 将源程序输入计算机，打入命令

```
vi example. c
```

vi 是命令驱动编辑程序，它包含丰富的编辑命令，详细内容请参看有关手册。

2. 编译

源程序编辑好后，就可调用 C 语言的编译程序获得相应的目标程序。打入命令

```
cc example. c
```

如果源程序有语法错误，编译系统会输出出错信息，告诉用户错误发生的位置和原因。这时需重新使用编辑程序，修改源程序，然后再编译。如此反复直至编译没有发现错误，得到正确的目标程序。

这里需特别指出，编译发现的错误位置通常在真正错误处之后。如程序中一个变量未定义，直到引用这个变量时才能发现错误；一个语句之后少了分号，只能在下一个语句开始处才能发现等。

编译程序最终产生与源程序等价的机器指令程序，称作目标程序。目标程序的文件名通常与源程序同名，而后缀为“o”。

3. 连接

连接是将目标程序和库函数或其他目标程序连接成可执行的程序文件。在 UNIX 系统下，简单程序的连接可由 C 语言的编译程序自动完成。cc 编译命令中，缺省的执行文件名为 a. out。如果用户指定执行文件名，譬如说 example. exe，则打入的编译命令为

```
cc -O example. exe example. c
```

其意思是对源程序 example. c 进行编译，产生的执行文件名为 example. exe。cc 编译命令可带许多任选参数，如上面的“-O example. exe”。详细内容请参看 UNIX 的 C 编译手册。值得一提的是，为组织大型 C 程序，UNIX 有个 make 命令，能在 makefile 中详细列出由众多文件组装成大程序，编译和连接的方法。有关内容请参见 UNIX 使用手册。

4. 执行

有了执行文件后，打入执行文件名，系统能自动读出执行文件到内存，并执行。如

a.out (编译时，用户未指定执行文件名)

或 example.exe (编译时，用户指定的执行文件名)

二、在 TURBO C 环境下编译运行 C 程序

TURBO C 为编辑、编译、运行 C 程序提供了一个开发环境。这里只介绍其中最简单的编辑、编译、运行 C 程序的方法。

1. 进入 TURBO C 程序开发环境

打入命令

```
tc
```

进入 TURBO C 程序开发环境，在屏幕顶部出现一行主菜单：

```
File Edit Run Compile Project Options Debug Break/watch
```

用键盘上的键“←”和“→”移动屏幕上的光标。当光标位于某个命令字上时，打回车键就表示请求执行该命令。开始时，光标位于“File”上，表示对有关文件类的操作。按回车键，在“File”下会出现一个子菜单窗口，有多项命令可供选择。用键“↑”和“↓”进行选择，将光标移到需要的命令处上。如将光标移到“Load”（或按 F3 功能键），按回车键，屏幕又出现一个小窗口，请指定装入的文件名。如输入 example.c。如果原来没有该文件，则建立这个文件；如已有这个文件，则该文件被读入并显示在屏幕上，并自动转入编辑(Edit)状态。

2. 编辑

根据步骤 1 进入 TURBO C 系统和装入源程序后，利用直观的全屏幕编辑方法编辑修改源程序。修改结束后，按“F2”功能键将源程序写到文件中。

使用命令

```
tc 文件名
```

能合并步骤 1 和步骤 2，直接进入编辑状态。

3. 编译

按“F9”功能键即可进行编译和连接。如程序有错，按任何一个键，屏幕显示源程序，光标停留在发现错误处。在屏幕下半部分显示发现错误处的程序行号和错误原因。按“F6”功能键再次进入编辑状态。修改完后，按“F9”功能键又进行编译。直至获得执行文件。

4. 执行

按“Ctrl”和“F9”两键（同时按），意指编译并运行，若有最新的执行文件，则直接执行。程序执行结束后，系统又自动回到原来状态。这时，按“Alt”和“F5”两键就能在屏幕上看到程序输出结果。如果运行结果不正确，又需修改源程序。如希望回到主菜单选择其它命令，可按“F10”功能键。在退出 TURBO C 程序开发环境后（见 5.），键入执行文件名称，也能执行程序。

5. 退出 TURBO C 程序开发环境

按“Alt”和 x 两键，使系统回到操作系统等待用户键入命令状态。

因 TURBO C 有程序开发环境，进入系统后，用户就不再需要打命令，只是简单的命令菜单选择。经上机操作实践，很快就可掌握各种命令的意义和使用方法。