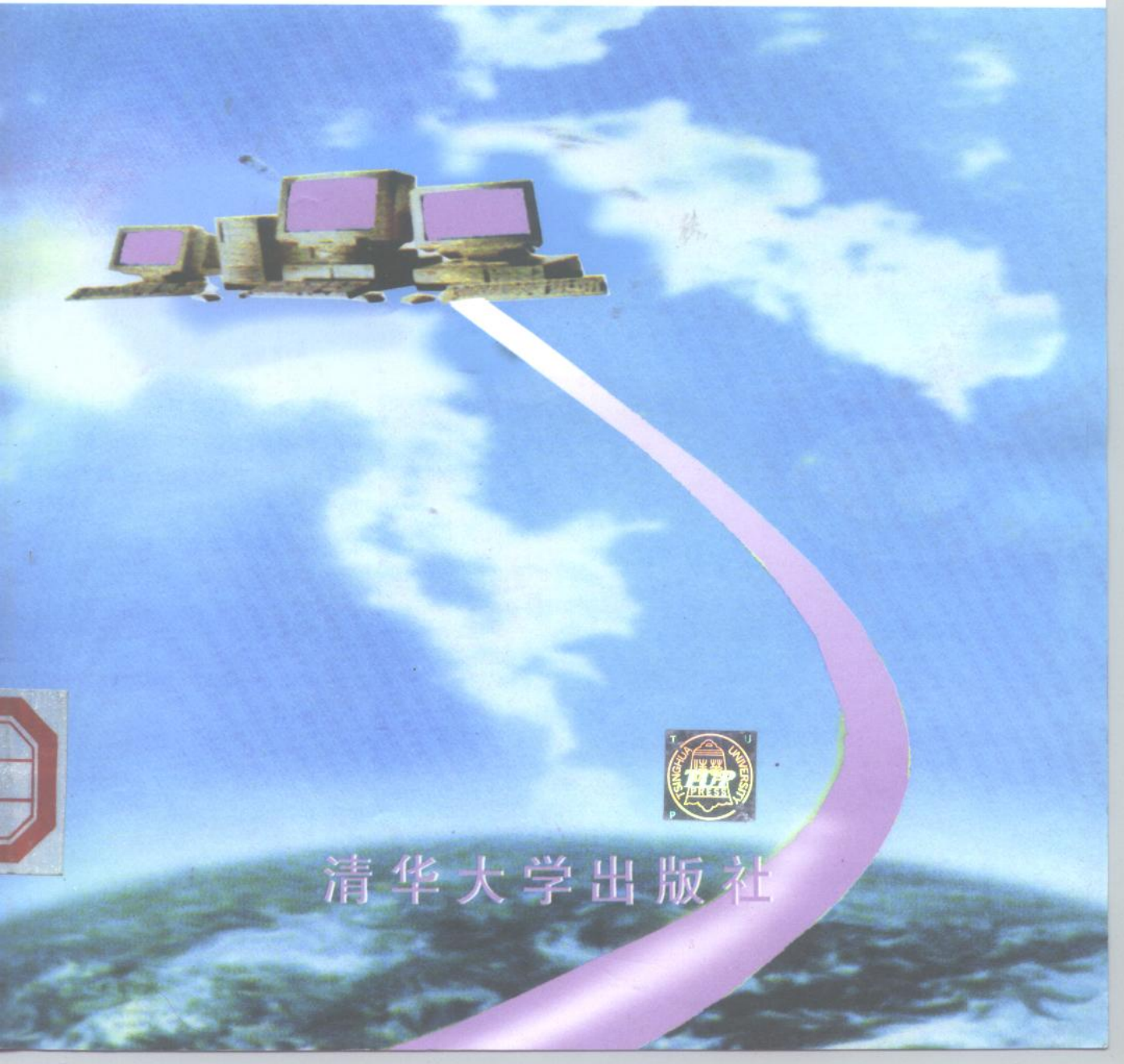
计算机专业大专系列教材

C语言程序设计

李大友 主编



清华大学出版社

计算机专业大专系列教材

C语言程序设计

李大友　主编

清华大学出版社

(京)新登字 158 号

JSJ45/5

内 容 简 介

C 语言是一种通用的程序设计语言，它不仅可以编写应用软件，而且特别适合于编写系统软件。

本书由 12 章组成，按照循序渐进的原则，逐步地介绍 C 语言中的基本概念和语法规则。同时，通过典型的例题分析，着重强调了利用 C 语言进行程序设计的方法。

本书是作者根据多年的教学经验编写而成的，在内容编排上尽量体现出易学的特点，在文字叙述上条理清晰、简洁，便于读者阅读。每章后附有习题，书末附有各章的习题解答，供读者学习使用，以利于全面地、系统地理解和掌握 C 语言程序设计。

本书可以作为计算机专业的大专教材使用，也可供非计算机专业作为本科教材使用。

图书在版编目(CIP)数据

C 语言程序设计/李大友主编. —北京：清华大学出版社，1999
计算机专业大专系列教材
ISBN 7-302-03103-7

Ⅰ.C… Ⅱ.李… Ⅲ.C 语言-程序设计-高等学校：专业学校-教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(1999)第 25310 号

出版者：清华大学出版社(北京清华大学学研楼，邮编 100084)
http:// www.tup.tsinghua.edu.cn
印刷者：大中印刷厂
发行者：新华书店总店北京发行所
开　本：787×1092 1/16　印张：14.5　字数：338 千字
版　次：1999 年 9 月第 1 版　2000 年 1 月第 2 次印刷
书　号：ISBN 7-302-03103-7/TP・2004
印　数：8001～19000
定　价：15.50 元

序　言

为什么要组织编写这套计算机专业大专使用的教材？根据什么来组织这套教材？这套教材的特点是什么？它能起到哪些作用？这就是我们在这篇序言中，要回答，也必须回答的问题。

计算机专业大专教育发展非常迅速，它反映了社会对这个层次人材的需求，在数量上已经超过了对本科人材的需求。大专这个层次有自己的特殊性，时间只有三年，要学习的内容很多，怎样精选教学内容，就成为十分重要的问题；他们又不同于中专层次，要求既有相当坚实的理论基础，又要运用理论解决实际问题，因此，如何处理好理论和实践的关系就十分重要。

大专这个层次人材的重要性是不言而喻的。但是，在培养大专这个层次人材的过程中，突出的矛盾之一，就是缺乏合适的大专教材。目前，不是用本科教材代用，就是很难及时获得所需教材。这就是组织这套专为大专使用的教材的起因。

那么，我们组织编写这套教材以什么为依据呢？中国计算机学会教育委员会与全国高等学校计算机研究会联合推荐的《计算机学科教学计划 1993》是我们这次组织编写这套教材的主要依据。

“93 计划”所提供的指导思想和学科内容不仅适合大学本科，也适合于大专的需要。

“93 计划”明确规定了计划实施的目标：1. 要为“计算机学科”的毕业生提供一个广泛坚实的基础；2. 在培养人材的过程中，必须反映培养目标的差异；3. 要为学生毕业后，进一步学习新的知识和迎接新的工作挑战，做好理论和实践上的准备；4. 要学生能够把在校学到的知识，用到解决实际问题的过程中去。

在学科内容方面“93 计划”概括了九个科目领域。九个科目领域组成《计算机学科》的主科目。每个科目领域都有重要的理论基础、重要的抽象（实验科学）、重要的设计和实现的成就。

这九个科目领域作为教学计划的公共要求，它们是：算法与数据结构、计算机体系结构、人工智能与机器人学、数据库与信息检索、人—机通信、数值与符号计算、操作系统、程序设计语言、软件方法学和工程。

我们根据上述指导思想和学科要求精选了十三种课，作为大专用的主干教材。它们是：《数据结构》、《数字电路逻辑设计》、《计算机组成原理》、《微型计算机原理》、《微型计算机接口技术》、《计算机网络》、《数据库原理及应用》、《操作系统基础》、《汇编语言程序设计》、《C 语言程序设计》、《软件工程概论》、《微机系统应用基础》和《离散数学》。

这十三种教材大体上反映了除人工智能与机器人学和数值与符号计算之外的全部要求，足以满足大专主干课程教学的需求。

这套教材我们都是聘请大专院校有丰富教学实践经验的工作在第一线的专家、教授编写的。在编写过程中，充分考虑了大专的特点，在选材上贯彻少而精的原则，在处理上贯

彻理论密切联系实际的原则。力求深入浅出，便于教学。并且在主要章节后均附有适量的习题。

这套教材适合于计算机专业大专生使用，也可供非计算机专业的本科生使用。

丛书主编　李大友

1996.6

前　言

C 语言是一种短小精悍的计算机高级程序设计语言，它是根据结构化程序设计原则设计并实现的。C 语言具有丰富的数据类型，它为结构化程序设计提供了各种数据结构和控制结构，它能够实现汇编语言中的大部分功能。同时，用 C 语言编写的程序具有特别好的可移植性。

尽管当初 C 语言是为编写 UNIX 操作系统而设计的，但它并不依赖于 UNIX 操作系统，目前 C 语言能在多种操作系统环境下运行，并且已经在广泛的领域里得到了应用，是目前国际上应用最为普及的高级程序设计语言之一。

多年来，作者一直从事 C 语言的教学工作，同时也利用 C 语言开发实际的课题，本书是作者根据多年的教学经验和应用 C 语言的体会写成的。本书的主要特点可归纳如下：

(1) 按照循序渐进的原则，逐步引出 C 语言中的基本概念。

(2) 在文字叙述上力求条理清晰、简洁，以利于读者阅读。

(3) 在讲解 C 语言中的基本概念时，除了阐述理论之外，还通过典型的例题，着重强调了基本概念在程序设计中的应用，以利于读者理解和掌握。

(4) 本书的重点是 C 语言的使用，书中没有深奥的理论和算法，在例题中出现的每一个算法，都给出了比较详细的解释。因此，特别适合于初学者和自学者使用。

(5) 本书的每一章中都包括“应用举例”一节，其中的例题包含了本章讲解的主要内容，有些例题还具有一定的难度，通过阅读和分析这些例题，使读者对本章内容的应用有一个全面的了解。

(6) 每章的最后都附有一定量的习题，其中包括程序分析题和编程题，这些习题对于读者巩固已学习的内容大有益处。在附录 E 中给出了各章习题的详细解答，可供读者参考。

作者认为，要学好 C 语言，除了掌握 C 语言的基本理论之外，还必须加强实践环节。本书中的所有例题都在微机(使用 Microsoft C)上调试通过，希望读者边学习边上机实践，这样不仅可以加快学习进度，也能提高学习效率。

本书由李大友教授主编，由李盘林教授、陈宪福和王旭老师编写，全书由李大友教授统稿和审定。文稿编辑录入由张华女士协助完成。

由于作者水平有限，书中一定有不少缺点和错误，敬请有关教师、计算机工作者和广大读者批评指正。

编　者

目　录

第1章 绪 言

C语言是目前世界上最为流行的计算机高级程序设计语言之一。它设计精巧，功能齐全，既适合于编写应用软件，又特别适合于编写系统软件。

C语言是在1972年至1973年间由D. M. Ritchie在贝尔实验室为描述和实现UNIX操作系统设计和实现的。在此以前像UNIX操作系统这样的系统软件，一般都是利用汇编语言这种低级语言来编写的，自C语言开发成功以来，使利用高级语言编写系统软件成为可能。UNIX操作系统源代码的90%以上是由C语言编写的。UNIX操作系统的一些主要特点，如便于理解、易于修改及具有良好的可移植性等，在一定程度上都受益于C语言。所以，UNIX操作系统的成功与C语言是密不可分的。

最初的C语言依赖于UNIX操作系统环境，随着C语言的不断发展及应用的普及，目前，C语言已经能够在多种操作系统，如UNIX，DOS等环境下运行。实用的C语言编译系统种类繁多，适用于IBM-PC机运行的有Microsoft C (MSC)，Turbo C (TC)和Lattice C (LC)等。

1.1 C语言的特点

C语言能够成为目前应用最广泛的高级程序设计语言之一，完全是由其语言特点决定的。C语言的特点可大致归纳如下：

(1) C语言短小精悍，基本组成部分紧凑、简洁。C语言一共有32个标准的关键字、45个标准的运算符以及9种控制语句，语言的组成精练、简洁，使用方便、灵活。

(2) C语言运算符丰富，表达能力强。C语言具有“高级语言”和“低级语言”的双重特点，其运算符包含的内容广泛，所生成的表达式简炼、灵活，有利于提高编译效率和目标代码的质量。

(3) C语言数据结构丰富，结构化好。C语言提供了编写结构化程序所需要的各种数据结构和控制结构，这些丰富的数据结构和控制结构以及以函数调用为主的程序设计风格，保证了利用C语言所编写的程序能够具有良好的结构化。

(4) C语言提供了某些接近汇编语言的功能，有利于编写系统软件。C语言提供的一些运算和操作，能够实现汇编语言的一些功能，如它可以直接访问物理地址，并能进行二进制位运算等，这为编写系统软件提供了方便条件。

(5) C语言程序所生成的目标代码质量高。C语言程序所生成的目标代码的效率仅比用汇编语言描述同一个问题低20%左右。因此，用C语言编写的程序执行效率高。

(6) C语言程序可移植性好。在C语言所提供的语句中，没有直接依赖于硬件的语句。与硬件有关的操作，如数据的输入、输出等都是通过调用系统提供的库函数来实现的，而这些库函数本身并不是C语言的组成部分。因此，用C语言编写的程序能够很容易地

从一种计算机环境移植到另一种计算机环境中。

当然，C 语言本身也有其弱点：一是运算符的优先级较多，不容易记忆；二是由于 C 语言的语法限制不太严格，这在增强了程序设计的灵活性的同时，在一定程度上也降低了某些安全性，这对程序设计人员提出了更高的要求。

总之，由于 C 语言的上述特点，使得 C 语言越来越受到程序设计人员的重视，并且已经在广泛的领域里得到了应用。

1.2 C 语言程序的开发过程

开发一个 C 语言程序的基本过程如图 1.1 所示。

1. 编辑

选择适当的编辑程序，将 C 语言源程序通过键盘输入到计算机中，并以文件的形式存入到磁盘中。在 DOS 系统下，经过编辑后得到的源程序文件都是以.C 为其文件扩展名。

2. 编译

通过编辑程序将源程序输入到计算机后，需要经过 C 语言编译器将其生成目标程序。在对源程序的编译过程中，可能会发现程序中的一些语法错误，这时就需要重新利用编辑程序来修改源程序，然后再重新编译。在 DOS 系统下，经过编译后得到的目标文件都是以.OBJ 为其文件扩展名。

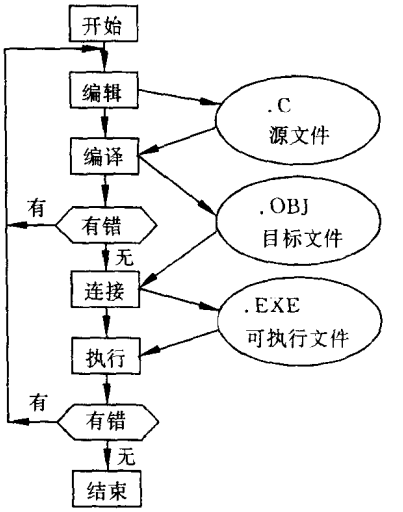


图 1.1

3. 连接

经过编译后生成的目标文件是不能直接执行的，它需要经过连接之后才能生成可执行的代码。在 DOS 系统下，连接后所得到的可执行文件都是以.EXE 为其文件扩展名。

4. 执行

经过编译、连接之后，源程序文件就可以生成可执行的文件，这时就可以执行了。在 DOS 系统下，只要键入可执行的文件名，并按“回车”键后，就可执行文件了。

现在有许多厂家都推出一种集成环境来处理 C 语言程序。如 Turbo C，在这种集成环境下，对程序的编辑、编译和连接等操作，都可以在一个窗口下进行，使用起来非常方便。

在本书附录 C 中简单介绍了 Turbo C 集成开发环境的使用方法，感兴趣的读者可以参考使用。

1.3 简单的 C 语言程序

这一节将介绍几个简单的 C 语言程序，并对其基本语法成分进行相应的说明，以便使读者对 C 语言程序有一个概括的了解。

例 1.1 编写一个 C 语言程序，显示字符串“Hello!”。

```
#include "stdio.h"
  main( )
{
  printf("Hello! \n");
}
```

上述程序是一个简单而完整的 C 语言程序，经过编辑、编译和连接后，其执行结果是在屏幕的当前光标位置处显示如下字符串：

Hello!

下面对上述程序进行说明：

(1) 一个 C 语言程序可以由多个函数组成，但任何一个完整的 C 语言程序，都必须包含一个且只能包含一个名为 main 的函数，程序总是从 main 函数开始执行。

(2) 由左右花括号括起来的部分是函数体，函数体中的语句将实现程序的预定功能。在本例中，main 函数的函数体中只有 printf 一个语句，它的功能是进行格式化输出(显示)，即将字符串“Hello! \n”显示在终端屏幕上。其中，字符串中的字符“\”和“n”合起来，表示一个“换行”字符，在“换行”字符后输出的任何字符，将被显示在屏幕的下一行上。

(3) C 语言中的每个基本语句，都是以“;”结束的。

(4) C 语言程序的书写格式比较自由，没有固定的格式要求，在一行内，既可以写一个语句，也可以写多个语句。为了提高程序的可读性，往往根据语句的从属关系，以缩进书写的形式体现出语句的层次性。

(5) #include 语句是编译预处理语句，其作用是将由双引号或尖括号括起来的文件中的内容，读入到该语句的位置处。在使用 C 语言输入、输出库函数时，一般需要使用 #include语句将"stdio.h"文件包含到源文件中。有关 #include 语句的作用及其使用方法，将在“编译预处理”一章中作详细介绍。

例 1.2 从键盘输入两个整数，并将这两个整数之和显示出来。

```
#include "stdio.h"
main( )              /* 计算两个整数之和 */
{
  int x,y,z;
  scanf("%d%d",&x,&y);          /* 读入两个整数，存入变量 x 和 y 中 */
  z=x+y;
  printf("The sum of %d and %d is %d",x,y,z);
}
```

上述程序经过编辑、编译和连接后，执行情况如下：

当程序执行到 scanf 语句时，将等待用户输入两个整型数据后再继续执行，假如用户输入 10 和 20(此时，x 的值为 10，y 的值为 20)，则程序将显示如下信息：

The sum of 10 and 20 is 30

下面，对上述程序进行说明：

(1) 程序中由/*和*/括起来的内容是程序的注释部分，它是为增加程序的可读性而设置的。注释部分对程序的编译过程和执行结果没有任何影响。

(2) C 语言中的所有变量都必须定义为某种数据类型。同时，必须遵循“先定义、后使用”的原则，如语句：

```
int x,y,z;
```

定义了 x,y,z 是 3 个整型变量，以后就可以使用这 3 个变量来存放整型数据。

(3) 一个 C 语言程序可以由多个函数组成，通过函数之间的相互调用实现相应的功能。程序中所使用的函数，既可以是系统提供的库函数，也可以是用户根据需要自己定义的函数。如上述 main 函数中调用的 scanf 函数和 printf 函数，就是系统提供的库函数。这些函数不需要用户自己定义，在需要时，只要按照指定的格式进行调用即可。C 语言编译系统提供的库函数非常丰富，特别是与硬件打交道的部分，很多工作只要调用库函数就可以实现。

(4) 程序中调用的 scanf 函数的作用是进行格式化输入。其中由圆括号括起的部分是函数参数部分，不同的函数需要不同的参数。scanf 函数中的参数主要包括两部分内容：一是“格式控制”部分，它用于对输入数据的格式进行说明；二是“地址表”部分(本书中出现的表的概念，如地址表、输出表等，是指用逗号分隔的有限个元素序列)，它使用的是存放输入数据的变量的地址。

程序中调用的 printf 函数的作用是进行格式化输出，其参数也包括两部分内容：一是“格式控制”部分，用于对输出数据的格式进行说明；二是“输出表列”部分，它使用的是存放输出数据的变量本身。有关数据的输入、输出以及函数的调用形式，将在以后做详细的介绍。

习　题

1.1 简述 C 语言的主要特点及其用途。

1.2 参照本章例题，编写一个 C 语言程序，用于显示以下信息：

```
Happy New Year!
```

1.3 参照本章例题，分析下面的 C 语言程序，并给出运行结果。

```
#include "stdio.h"
main( )
{
  int a,b,c;
  a=100;
  b=20;
  c=a-b;
  printf("sum=%d",c);
}
```

1.4 参照本章例题，编写一个 C 语言程序，用于显示以下信息：

```
##########################
  Wonderful!
##########################
```

1.5 参照本章例题，分析下面的C语言程序，并给出模拟运行结果。

```
#include "stdio.h"
    main( )
{
  int x,y,z;
  scanf("%d%d",&x,&y);
  z=x/y;
  printf("Result=%d",z);
}
```

第 2 章　数据类型、运算符及其表达式

C 语言程序中所用到的每一个常量、变量及函数等都是程序的基本操作对象，它们都隐式地或显式地与一种数据类型相联系。每种数据类型都表明了它的可能取值范围及能在其上所进行的运算。

C 语言中的数据类型，大体上可划分为基本的数据类型和导出的数据类型两种，基本数据类型主要包括整型、字符型和单、双精度浮点型等；导出数据类型是在基本数据类型的基础上产生的，其中包括数组、结构等。

本章主要讨论 C 语言中变量名及其常量的构成形式，说明 C 语言中的几种基本数据类型、算术运算符及构成算术表达式的一些基本规则。

2.1　常量和变量

1. 常量

常量又称常数，是指在程序运行过程中其值不能被改变的量。常量也分为不同的类型，这是由常量本身隐含决定的，下面将详细介绍。为了增加程序的可读性，可以用一个名字(字符序列)代表一个常量，此时的常量称为符号常量。有关符号常量的使用，将在“编译预处理”一章中详细介绍。

2. 变量

变量是指在程序运行过程中其值可以被改变的量。变量可分为不同的类型，不同类型的变量在内存中占用不同的存储单元，以便用来存放相应变量的值。

在计算机语言中，经常用到标识符的概念。所谓标识符，是指用来标识程序中用到变量名、函数名、类型名、数组名、文件名以及符号常量名等的有效字符序列。

需要说明的是，保留字又称关键字，也是 C 语言中的一种标识符，它用来命名 C 语言程序中的语句、数据类型和变量属性等。每个保留字都有固定的含义，不能另作其他用途。

在 C 语言中，标识符的命名规则是：由字母(大、小写皆可)、数字及下划线组成，且第一个字符必须是字母或下划线。

由上述标识符的命名规则可知，下面的变量名是合法的：

year，Day，ATOK，x1，_CWS，_change，_to

而下面的变量名是不合法的：

#123，.CUM，$100，1996Y，1－2－3－

在 C 语言中，大写字母和小写字母是有区别的，即作为不同的字母来看待。如变量 RAN，Ran 和 ran 分别表示 3 个不同的变量，这一点同其他高级语言是有区别的，应引起注意。

组成变量名(标识符)的有效字符数随 C 语言的编译系统而定。有的编译系统允许使

用长达31个字符的变量名，而有的编译系统只取变量名的前8个字符作为有效字符，后面的字符无效，不被识别。这样，只要变量名的前8个字符相同，就被认为是同一个变量。因此，在进行程序设计之前，应首先了解所使用的编译系统中对变量名长度的规定，以免造成变量使用上的混乱。

2.2 基本数据类型及其常量

在C语言中，最基本的数据类型只有四种，它们分别由如下标识符进行定义：

int　　整型
char　　字符型
float　　单精度浮点型
double　　双精度浮点型

C语言规定，对程序中用到的所有变量，都必须先定义，后使用。在定义变量时，不能把C语言中具有固定含义的保留字（如int，char等）作为变量名；同时，同一个函数内所定义的变量不能同名。

2.2.1 整型变量及其常量

1. 整型变量

整型变量可用来存放整型数据（即不带小数点的数）。其定义方式如下：

```
int i1,i2;
```

其中i1和i2即定义为整型变量。

2. 整型常量

在C语言中，整型常量可以用三种数制表示：

（1）十进制整型常量：如，250，－12等，其每个数字位可以是0～9。

（2）十六进制整型常量：如果整型常量以0x或0X开头，那么这就是用十六进制形式表示的整型常量。如，十进制数的128，用十六进制表示为0x80或0X80，其每个数字位可以是0～9，A～F。

（3）八进制表示的整型常量：如果整型常量的最高位为0，那么它就是以八进制形式表示的整型常量。如，十进制数的128，用八进制表示为0200。需要注意的是，八进制数中的每个数字位必须是0～7。

2.2.2 浮点型变量及其常量

1. 浮点型变量

在C语言中，把带有小数点的数称为浮点数，也可以称为实型数。

浮点型变量又称为实型变量，按其能够表示的数的精度，又分为单精度浮点型变量和双精度浮点型变量。

单精度浮点型变量的定义方式为

```
float f1,f2;
```