高等教育规划教材

C/C++程序设计

范翠香　刘　辉　编著

提供电子教案和习题解答

下载网址 http://www.cmpedu.com

机械工业出版社
CHINA MACHINE PRESS

高等教育规划教材

C/C++程序设计

范翠香　刘　辉　编著
胡忻利　主审

机械工业出版社

C 语言作为一门最通用的程序设计语言，学习并掌握 C 语言是每一位计算机相关专业学生必须掌握的一个专业技能，也几乎是每一个理工科或者其他专业的学生必须具备的基本功之一。

本书以程序设计思想为主导，详细介绍了程序设计的基本知识、C 语言基本知识、结构化程序设计方法、数组、指针、函数、结构体、编译预处理、文件操作和 C 语言程序调试技能，同时对于面向对象程序设计的基本概念也进行了介绍。本书内容翔实、知识体系合理，知识引入深入浅出，并提供大量实用例题以及丰富多样的习题，方便读者使用。

本书可作为高等本科院校计算机科学和电子与信息工程等相关专业的程序设计基础课程的教材，也可作为计算机与电子信息相关专业的程序设计基础学习参考教材。由于本书深入浅出知识引入方法，故本书也特别适合自学者使用。

本书配有电子教案和习题解答，需要的教师可登录 www. cmpedu. com 免费注册，审核通过后下载，或联系编辑索取（QQ：2966938356，电话：010 - 88379739）。

图书在版编目(CIP)数据

C/C ++ 程序设计/范翠香，刘辉编著 . —北京：机械工业出版社，2017. 7
高等教育规划教材
ISBN 978-7-111-56730-1

Ⅰ. ①C… Ⅱ. ①范… ②刘… Ⅲ. ①C 语言 - 程序设计 - 高等学校 - 教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2017)第 099974 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
策划编辑：和庆娣 责任编辑：和庆娣
责任校对：张艳霞 责任印制：李 昂
三河市宏达印刷有限公司印刷
2017 年 6 月第 1 版 · 第 1 次印刷
184 mm × 260 mm · 17. 5 印张 · 427 千字
0001 - 3000 册
标准书号：ISBN 978-7-111-56730-1
定价：45. 00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
服务咨询热线：(010)88379833
读者购书热线：(010)88379649

网络服务
机 工 官 网：www. cmpbook. com
机 工 官 博：weibo. com/cmp1952
教育服务网：www. cmpedu. com
金 书 网：www. golden - book. com

前　言

C 语言从产生到现在，已经成为最重要和最流行的编程语言之一。在各种流行编程语言中，都能看到 C 语言的影子，如 Java、C#的语法与 C 语言基本相同。学习、掌握 C 语言是每一个计算机技术人员的基本功之一。同时，C 语言作为一门通用的语言，几乎是每一个理工科或者其他专业的学生都要学习的语言。

本书在编写过程中，注重了知识内容的体系结构，力求做到内容翔实且突出重难点，如将指针放在函数之前，确保在函数的应用中可以全方位引入指针，如按地址传递参数的不同实现形式、函数返回地址等。特别地，将 C 语言集成环境以及 C 程序的各种连编和程序调试方法单独列为一章，增加了模块化的、由多个源文件组成的 C 程序的编译调试方法，这部分内容在目前已出版的 C 语言教材中比较少见。

本书在 C 语言基础上增加了面向对象的内容，考虑到许多院校专门开设有面向对象的程序设计课程如 Java 等，在这些课程中会对面向对象的知识进行详细介绍，故本书对于面向对象只介绍了基本概念和基本思想，重点介绍了面向对象的抽象和封装这两个基本特征。读者不仅可以初步了解面向过程的程序设计方法与面向对象的程序设计方法的不同之处，同时通过类中对成员函数的设计也可以进一步加深对函数的理解和应用。

本书共分 11 章，第 1 章介绍了程序设计基础、C 语言的发展及特点；第 2 章介绍了 C 语言的数据类型、基本运算符及表达式、各种不同类型数据的输入与输出；第 3 章介绍了结构化程序设计的 3 种基本结构；第 4 章介绍了数组及其应用；第 5 章介绍了指针及其应用；第 6 章介绍了函数及其应用；第 7 章介绍了编译预处理和位运算；第 8 章介绍了结构体类型、共用体类型和枚举类型；第 9 章介绍了数据文件的应用；第 10 章介绍了面向对象的程序设计基础知识；第 11 章介绍了 C 语言的集成环境与各种程序调试方法。

书中的每一章都提供了丰富且实用的例题，提供了较细致的算法分析图表，帮助读者理解并掌握基本算法及算法设计技巧。每一章后也配备了丰富的不同类型的习题。

本书中的程序代码均在 Visual C ++ 6.0 环境中调试通过。

本书由西安理工大学信息装备与控制工程学院范翠香、刘辉编著。范翠香老师编写了第 1、2、3、5、7、8、9、11 章和附录，刘辉老师编写了第 4、6、10 章。全书由范翠香、刘辉老师统稿，胡忭利老师主审。

由于作者水平有限，书中难免存在不妥和疏漏之处，恳请读者批评指正，谢谢！

编　者

目　录

第 1 章　程序设计基础及 C 语言概述

内容提要　本章主要介绍程序、程序设计以及算法等有关概念；介绍 C 语言的发展、特点和 C 语言源程序的组成。

目标与要求　理解程序和程序设计、算法的概念及特点；掌握算法的描述方法；了解 C 语言源程序的组成。

1.1　程序与程序设计语言

计算机语言是实现人机交互的有效工具。在计算机帮助用户解决某个问题前，用户首先要编写解决这个问题的程序，而程序的编写规范是基于某种特定计算机语言的。本节介绍程序、程序设计以及程序设计语言的基本概念。

1.1.1　程序和程序设计

1. 程序

为了让计算机完成特定的任务，人们事先编制的一组指令的有序集合，称为程序。而指令是计算机硬件能识别并直接执行的一个基本操作命令。

2. 程序设计

程序设计是指从人们分析实际问题开始到计算机给出正确结果的整个过程，如图 1-1 所示。

在这个过程中，建立数学模型和确定数据结构与算法是程序设计中最难的一步，需要有一定的数学基础和数值计算方法等知识。编写程序就是用某一种计算机语言将设计的算法过程描述出来。

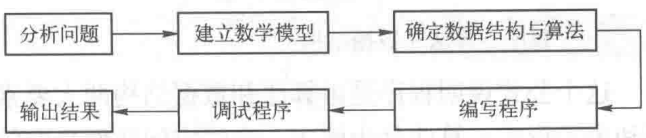

图 1-1　程序设计过程

1.1.2　计算机语言

1. 程序设计语言的概念

计算机语言是计算机能够识别的语言，是人与计算机进行信息交流的工具，也称为程序设计语言或编程语言。目前计算机语言有上百种，每一种计算机语言都有一套固定的符号和语法规则。人们要利用计算机解决问题，就必须用计算机语言来告诉计算机“做什么”和“怎样做”，这个过程就称为编程。

伴随着计算机技术的不断发展和变化，计算机语言也经过了几个发展阶段，从机器语言到汇编语言再到高级语言，计算机语言详细发展过程这里不再叙述。

2. 高级语言的处理过程

计算机能直接识别并执行的指令只能是二进制指令。用高级语言编写的程序称为高级语言源程序，计算机并不能直接执行它，必须将其翻译成机器能识别的语言形式（即二进制

程序，也称目标程序）。而这种具有翻译功能的系统程序称为编译程序，大多数高级语言都有自己的编译程序。

高级语言程序的处理方式分为解释方式和编译方式。

1）解释方式：执行源程序时，采用边翻译边执行方式，即翻译一句执行一句，不产生目标程序。早期的 Basic 语言采用的就是这种方式。

2）编译方式：执行源程序前，使用编译程序先将高级语言源程序翻译成机器语言程序（即扩展名为 obj 的二进制的目标文件）。这个文件也不能直接运行，还需要用连接程序将系统提供的库函数和头文件等连接到目标文件，生成一个扩展名为 exe 的可执行文件，此文件才可以独立运行。早期的 Pascal 语言和本书学习的 C 语言都是采用这种方式。

高级语言源程序编译、连接的过程如图 1-2 所示。

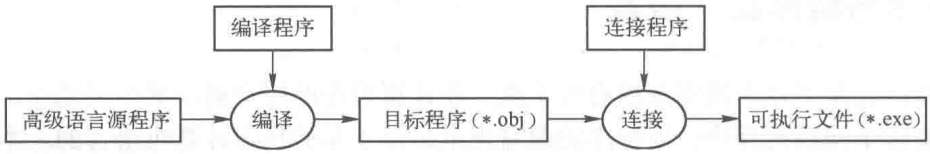

图 1-2　高级语言源程序编译、连接的过程

1.1.3　算法

1. 算法的概念

算法（Algorithm）一词源于算术（Algorism），即算术方法，是一个由已知推求未知的运算过程。后来，人们把进行某一工作的方法和步骤称为算法。

程序设计的关键步骤是算法设计。算法在很大程度上决定了程序的效能。著名的计算机科学家 Niklaus Wirth 曾提出：

```
程序 = 算法 + 数据结构
```

这个公式说明程序是由算法和数据结构两大要素构成的。其中，数据结构是指数据的组织和表示形式；算法是为解决一个特定问题而采取的确定且有限的步骤。

2. 算法的特点

算法有如下 5 个特点。

1）有穷性。一个算法应包含有限个操作步骤。在执行有限个操作之后，算法能正常结束，而且每一步都能在合理的时间范围内完成。

2）确定性。算法中的每一条指令必须有确定的含义，不能有二义性，对于相同的输入必须得出相同的执行结果。

3）可行性。算法中每一步操作，都必须是计算机能识别且能通过执行基本运算完成的。

4）有零个或多个输入。大多数情况下，算法执行的开始都会有原始数据提供，这里的输入并不一定指键盘输入，也可能从文件中读取。

5）有一个或多个输出。算法的运行结果必须以某种形式反馈给用户。当然这里的输出也不一定指屏幕输出或打印，也可以将结果输出到一个磁盘文件中。

3. 算法的描述方法

算法可以用多种方法来描述，最常用的是伪代码、流程图和 N－S 流程图。

（1）伪代码

伪代码是用接近人类自然语言（与高级语言类似），但又不受严格的语法限制的一种算法描述方式。

（2）流程图

流程图是用一些几何框图和流程线表示算法的执行过程。这种方式形象直观，易于理解，被普遍采用，但占用篇幅大。流程图常用符号如图 1-3 所示。

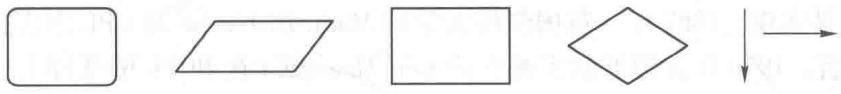

图 1-3　流程图常用符号

a）起止框　b）输入/输出框　c）处理框　d）判断框　e）流程线

（3）N-S 流程图

N-S 流程图是美国学者 I. Nassi 和 B. Shneiderman 在 1973 年提出的一种流程图形式，用两个学者名字的首字母命名。它去掉了流程线，每一步用一个矩形框来表示，把一个个矩形框按执行的顺序连接起来，使得算法只能从上到下执行。使用这种方法作图简单，描述占用面积小，因此很受欢迎。

【例 1-1】某门课程的总评成绩计算方法：总评成绩 = 平时成绩（占总评的 20%）+ 期末成绩（占总评的 80%）。

现已知某学生某门课程的平时成绩和期末成绩，编程计算该学生的总评成绩，并给出该课程是否通过。算法的 3 种描述方法如图 1-4 所示。

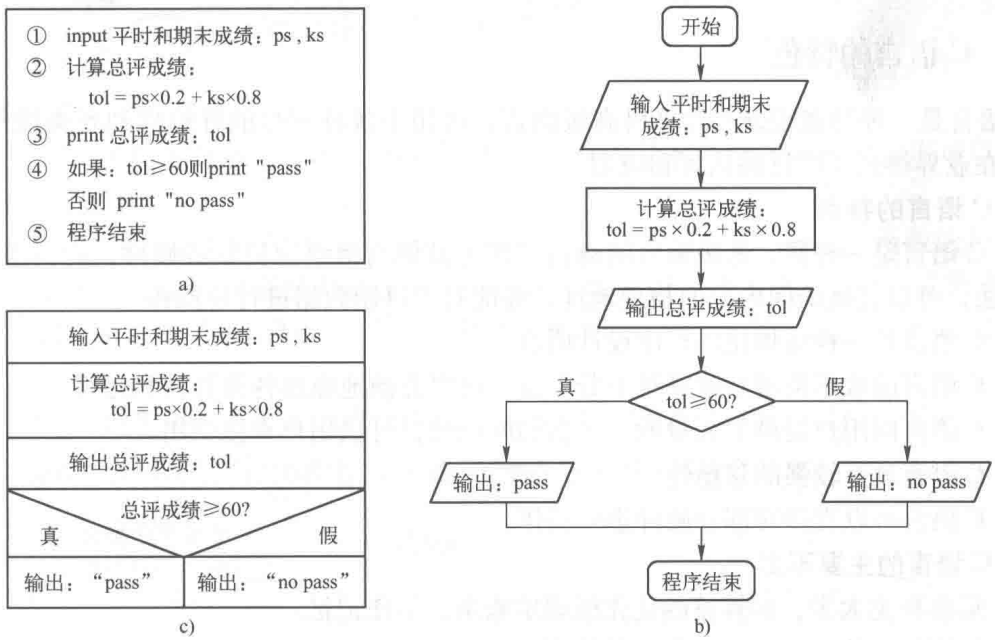

图 1-4　【例 1-1】的 3 种算法描述方法

a）伪代码　b）流程图　c）N-S 流程图

1.2　C 语言概述

目前计算机语言种类很多，有些语言有其特定的应用领域，例如，PHP 专门用来显示

网页；Perl 更适合文本处理；C 语言被广泛用于操作系统和编译器（所谓的系统编程）的开发。

1.2.1 C 语言的发展

C 语言最早的原型是 Algol 60。1963 年英国剑桥大学在 Algol 60 的基础上推出了 CPL（Combined Programming Language）语言。CPL 语言较 Algol 60 更接近硬件一些，不足之处是规模过大，不易实现。1967 年，英国剑桥大学的 Matin Richardsc 对 CPL 语言进行简化，开发了 BCPL 语言。1970 年美国贝尔实验室的 Ken Thompson 在 BCPL 的基础上设计出了更简单且更接近硬件的 B 语言，并用 B 语言编写了 UNIX 操作系统。

由于 B 语言依赖于机器，过于简单，功能有限，又无数据类型，所以并没有流行起来。1972～1973 年间，贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言。C 语言既保持了 B 语言的特点（精炼、接近硬件），又克服了它的缺点（过于简单、无数据类型等）。1973 年，Ken Thompson 和 D. M. Ritchie 合作，把 UNIX 中 90%以上的代码用 C 语言改写。后来，C 语言进行了多次改进，1977 年出现了不依赖具体机器的 C 语言版本。随后，C 语言迅速普及，成为当今广泛使用的计算机语言之一。

目前流行的 C 语言版本都是以标准 C（ANSI C）为基础的，各种版本的基本部分相同，个别地方稍有差异，经常使用的有 Microsoft C 5.0、Turbo C 2.0 等版本。

随着面向对象编程技术的发展，又出现了 C++、Turbo C++、Visual C++等版本。C++是 C 语言的延伸，它是在 C 语言的基础上，增加了面向对象的概念，成为一种流行的面向对象的程序设计语言，功能更加强大。

1.2.2 C 语言的特色

C 语言是一种功能很强的计算机高级语言，可用于各种型号的计算机和各类操作系统，因此，在业界得到了广泛的认可和应用。

1. C 语言的特点

1）C 语言是一种高、低级兼容的语言。它比其他高级语言更接近硬件，比低级语言更接近算法，可以直接访问内存的物理地址，并能对二进制数据进行位操作。

2）C 语言是一种结构化的程序设计语言。

3）C 语言的数据类型和运算符十分丰富，可以方便地描述各种算法和运算。

4）C 语言向用户提供了大量的、丰富的库函数，可供用户直接调用。

5）C 语言具有较强的移植性。

6）C 语言可以直接对部分硬件进行操作。

2. C 语言的主要不足

1）运算种类太多，运算符的优先级规定繁杂，不便记忆。

2）数据类型检验太弱，转换比较随便。

3）没有数据边界自动检查功能，使用不太安全。

3. C 语言的源程序的结构特点

下面给出用两种实现方法编写的【例 1-1】的 C 语言源程序，希望能使大家对 C 程序有初步的认识。

实现方法 1：在 main( )函数中实现，程序如下。

```
#include <stdio.h>
void main()
{   int ps,ks;
    float tol;
    scanf("%d %d",&ps,&ks);
    tol=ps*0.2f+ks*0.8f;
    printf("tol=%0.1f\n",tol);
    if (tol>=60.0f)
            printf("pass");
    else
            printf("no pass");
}
```

实现方法2：使用两个函数实现，程序如下。

```
#include <stdio.h>
float computer_tol(int sc1,int sc2,float p1,float p2)   //p1、p2为两个成绩所占百分比
{   float tol;
    tol=sc1*p1+sc2*p2;
    return tol;
}
void main()
{   int ps,ks;
    float tol;
    scanf("%d%d",&ps,&ks);
    tol=computer_tol(ps,ks,0.2f,0.8f);    //调用函数computer_tol()计算总评成绩
    printf("tol=%0.1f\n",tol);
    if (tol>=60.0f)
            printf("pass");
    else
            printf("no pass");
}
```

C语言源程序的结构特点如下。

1）C语言源程序由函数组成，函数是构成C程序的基本单位。一个C程序可以由一个或多个不同的函数组成。

2）一个C程序中有且仅有一个main()函数（称为主函数）。main()函数的位置可放在程序的任何地方。C程序总是从main()函数开始执行，并且在main()函数中结束。main()函数可以调用其他函数，而其他函数不能调用main()函数，其他函数之间可以互相调用。

3）每一个函数均由两部分组成，即函数声明部分和函数体部分，其结构如下。

```
函数类型 函数名(形式参数表)   //函数声明部分(也称函数头)
{
    变量声明部分
    可执行语句部分          } 函数体
}
```

4）C语句使用“;”作为语句的结束符。

5）在C程序中，大小写字母代表不同的含义。

6）“//”后面直到所在行结束的内容均为注释内容，用于对前面的语句或程序的功能进行说明。注释内容不会被程序执行，对程序运行结果不产生影响。注释可放在程序中的任何地方，若需对一个语句或对程序的功能加以说明，通常用“//”；若注释的内容为一段而非一行时，用“/*　注释内容　*/”进行注释，其中“/*”和“*/”要成对出现，且

“/”和“*”之间不允许出现空格。

7）C 程序书写格式自由。可以一行写一条语句，也可以一行写多条语句，还可以一条语句分几行写。通常，一行写一条语句，便于阅读。

4. C 程序的编写风格

C 程序编写格式灵活，若从编写清晰、便于阅读理解、易于维护的角度出发，在编写程序时最好遵循以下规则。

1）一个说明或一个语句占一行。

2）用“{}”括起来的部分，通常表示程序的某一层次结构。“{}”一般与该结构语句的第一个字母对齐。

3）用分层缩进的格式显示嵌套的层次结构，可以使层次看起来更加清晰，并增加程序的可读性。示例如下。

```
int main()
{
    int x[10],i;
    int max;
    for(i=0;i<10;i++)
        scanf("%d",&x[i]);
    max=x[0];
    for(i=1;i<10;i++)
        if (x[i]>max)
            max=x[i];
    printf("max=%d\n",max);
    return 0;
}
```

4）在程序中适当加注释，有助于阅读、理解程序。

良好的程序设计风格有助于程序的维护，在学习程序设计的初期，应注意培养好的编写风格，养成良好的习惯，减少错误，从而提高编程及调试程序的效率。

关于 C 程序的运行环境及程序调试方法详见第 11 章。

习题 1

一、选择题

1. 以下不是算法特点的是（　　）。

A. 确定性　　B. 有穷性　　C. 效率高　　D. 可行性

2. C 程序的基本组成单位是（　　）。

A. 语句　　B. 函数　　C. 程序　　D. main()函数

3. 以下叙述正确的是（　　）。

A. C 语言比其他语言高级

B. C 语言程序不用编译就可以直接运行

C. C 语言是结构化的程序设计语言，是面向过程的语言

D. C 语言是面向对象的程序设计语言

4. 关于 C 语言，下列叙述中正确的是（　　）。

A. 计算机能直接运行 C 语言源程序

B. C 语言源程序经编译后，生成的 obj 文件可以直接运行

C. C 语言源程序经编译、连接后，生成的 exe 文件可以直接运行

D. C 语言采用解释方式翻译源程序

5. 关于 C 程序，下列叙述中正确的是（　　）。

A. 程序是按照从上往下的顺序运行

B. 程序是从 main( ) 函数开始运行，并在 main( ) 函数中结束

C. C 程序中不区分大小写

D. 一个 C 程序只能有一个函数

二、填空题

1. 一个 C 程序有且只能有一个_____函数，程序执行从_____开始，程序结束也从_____结束。

2. C 语句以_____ 作为结束标志。一行可以书写_____语句。

3. 在 C 程序中，“/*”和“*/”之间的内容为_____，这部分内容计算机_____。

4. 上机运行一个 C 语言源程序必须经过_____ 、_____、_____和_____ 4 个步骤。

5. 请用伪代码写出求两个整数平方和的算法描述：_____ 。

6. 一个程序的好的书写应按_____格式。

第 2 章　数据类型、运算符及表达式

内容提要　用 C 语言编写程序解决一些实际问题时，首先遇到的就是各种数据的表示、存储、计算以及输入与输出。本章主要介绍 C 语言的各种数据类型、常量与变量、运算符和表达式以及数据的输入与输出。

目标与要求　了解数据类型的概念，掌握 C/C ++ 的基本数据类型；掌握常量、变量的概念及表示方法；重点掌握整型、实型、字符型数据的使用方法；掌握各种运算符的功能及优先级、结合方向；掌握表达式的书写格式和规则；掌握 putchar( )、getchar( )、printf( )和 scanf( )函数的基本使用方法。

2.1　数据类型

用户在处理不同问题时，会遇到各种不同表现形式的数据，这些不同表现形式的数据在计算机中会采用不同的存储方式，因而对其加工处理的方法也不相同。

2.1.1　C 语言的数据类型

计算机中用数据类型来表示数据的存储和加工处理方法的属性。

C 语言提供了丰富的数据类型，其分类如图 2-1 所示。最常用的是基本类型，基本数据类型及其说明如表 2-1 所示。

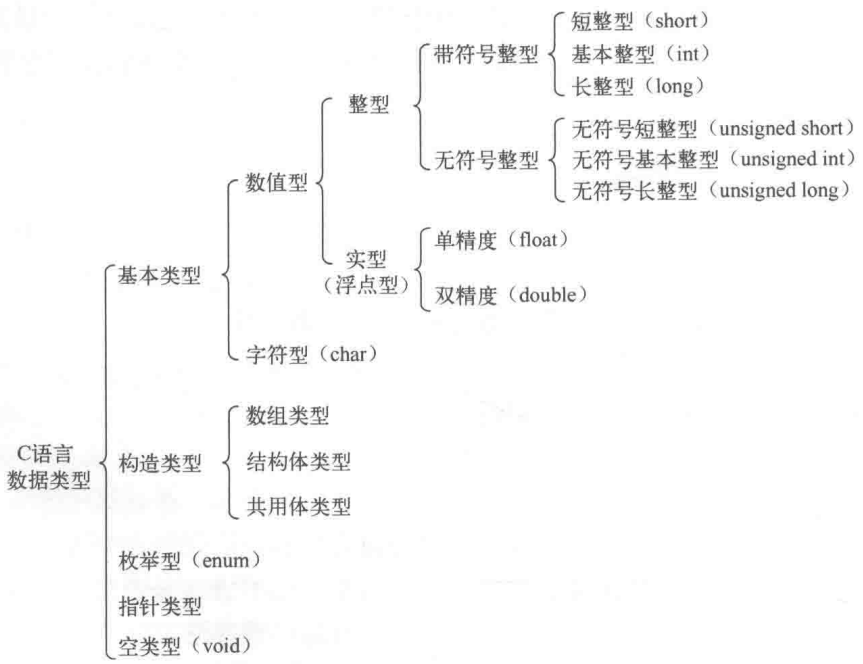

图 2-1　C 语言的数据类型

表 2-1　C 语言的基本数据类型及说明

数据类型及标识符		在 Turbo C 2.0 中		在 Visual C ++6.0 中	
		字节数	取值范围	字节数	取值范围
整型数据	有符号短整型（short）	2	（-32 768 ~ +32 767）	2	（-32 768 ~ +32 767）
	有符号整型（int）	2	（-32 768 ~ +32 767）	4	-2 147 483 648 ~2 147 483 647
	有符号长整型（long int）	4	-2 147 483 648 ~2 147 483 647	4	-2 147 483 648 ~2 147 483 647
	无符号短整型（unsigned short）	2	（0 ~65 535）	2	（0 ~65 535）
	无符号整型（unsigned int）	2	（0 ~65 535）	4	（0 ~4 294 967 295）
	无符号长整型（unsigned long）	4	（0 ~4 294 967 295）	4	（0 ~4 294 967 295）
实型数据	单精度（float）	4	-10^{38} ~10^{38}有效位：6 ~7 位		
	双精度（double）	8	-10^{308} ~10^{308}有效位：15 ~16 位		
	长双精度（long double）	8	-10^{308} ~10^{308}有效位：15 ~16 位		
字符型数据	字符型（char）	1	无符号数：0 ~255；有符号数：-128 ~ +127		
	字符串型	字符串型数据只有字符串常量，无字符串变量			

2.1.2　常量与变量

1. 常量

常量是指在程序的执行过程中其值不能被改变的量，分为直接常量和符号常量。

（1）直接常量

在程序中常数即为直接常量，简称常量。如：123、1.25、'*'、"abcd"等都是直接常量。

（2）符号常量

在 C 程序中，如果一个直接常量被反复多次使用，则可以定义一个符号（标识符）来代替它，我们称这个能代替该直接常量的符号（标识符）为符号常量。符号常量是一类特殊的常量，其值在程序的运行过程中是不能改变的。符号常量必须先定义后使用。定义的方法是在程序的开头用下述命令。

```
#define  符号常量名  常量
```

符号常量仍代表的是常量，一旦定义不能对其再重新赋值。习惯上符号常量名用大写字母表示。使用符号常量可以提高程序的可读性，减少输入工作量，实现“一改全改”。

例如，在处理一个问题时，当问题规模 N 不能完全确定时，则可以先将其定义为一个符号常量 N，并赋予一个比较大的值，在程序中，凡是引用到该问题规模时，均可用 N 代替。定义方法如下。

```
#define  N  100                    //定义符号常量 N 代表问题规模 100
```

用这个 N 代表所处理问题的规模 100，在程序中凡引用到这个数据规模的地方均用 N 代替。假设处理问题的数据规模改变为 10 000 时，用户不需修改程序中的任何代码，只需直接将符号常量 N 的定义值改为 10 000 即可。

2. 变量

（1）变量的定义

在程序的运行过程中，其值可以被改变的量，称为“变量”。在程序中，通常用变量来存储输入的数据、计算的一些中间值以及最终的程序处理结果。

（2）变量名和变量值

变量是用来存储数据的，C 语言规定，每个变量都必须有一个符合 C 语言标识符命名规则的名字，这个名字称为变量名。变量名代表了内存中分配给它的存储单元，在这些存储单元中，存放该变量的值。程序中通常通过变量名获取该变量的值。

图 2-2 形象表示了变量、变量的存储单元以及变量值三者之间的关系。其中，x 是变量名，矩形框表示变量 x 所占的存储单元，框内的数值 39 是变量 x 的值。

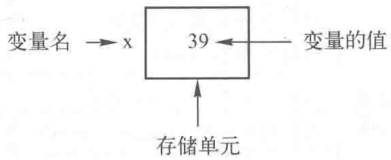

图 2-2　变量、变量的存储单元与变量的值之间的关系

（3）标识符命名规则

程序中所有用到的变量、函数、符号常量等都必须有其合法的名字，以示区别。而所有名字都必须符合 C 语言标识符的命名规则。

C 语言标识符的命名规则如下。

1）只能以字母或下画线开头。

2）在第一个字符的后面，只能是字母、数字或下画线 3 种字符，其他字符均不可以使用。

3）区分字母的大、小写，大小写字母代表不同含义。

4）长度没有限制，但不宜过长。

5）保留字不能作为标识符使用。C 语言中把用于描述变量的存储类型、数据类型以及程序流程控制等的字符序列称为“保留字”或“关键字”，这些关键字已经赋予了特定的含义和用途，不能再作为用户自定义的标识符使用，如标识符不能使用 int、while、if、switch 等。C 语言常见关键字见附录 A。

给变量命名时，尽量做到“见名知义”，通常用小写字母表示。例如，year、name、age、r、a_1 等。当名称中含有多个单词时，通常使用骆驼命名法，即从第二个单词开始，每个单词的第一个字母大写，如 userName、myFunction 等。

【例 2-1】 判断下列哪些字符序列可以作为合法的用户自定义标识符使用。

```
ab    _1    B1    c_2    x.c    int    xy1    b1    1.c    Float    a+b    -a
```

解： 依据上述标识符命名规则，合法的标识符有：

```
ab    _1    B1    c_2    xy1    b1    Float
```

（4）变量的类型

根据程序需要，一个变量的类型可以定义为 C 语言的任何一种数据类型。

当变量的类型声明后，编译系统会根据变量的数据类型为其在内存中分配相应数目且连续的存储单元，并且用该变量名标识该存储单元。

（5）变量的定义

C 语言规定，变量必须先定义后使用。变量的定义，就是根据程序功能的需要，确定程序中用到的变量名、变量的类型。编译时，系统会根据其声明类型分配给它相应数目的存储单元。

变量定义的格式如下。

```
[ 存储类型符 ]  类型符  变量名1,变量名2,…;
```

其中，类型符为变量的数据类型，可以是 C 中的任何一种数据类型。变量名必须遵循标识符的命名规则。用户可在一个数据类型后定义多个同类型变量，各变量名之间用逗号分