第 1 章

程序设计与 C 语言

自 1946 年世界上第一台数字计算机诞生以来，计算机技术得到了飞速发展，计算机系统已具有相当高的水准，但仍采用冯·诺依曼体系结构，即存储程序结构，这说明计算机的执行必须有程序控制，因此利用计算机解决问题，首先要解决程序设计问题。之后程序设计语言也得到了迅速发展，先后出现了机器语言、汇编语言和高级语言。

C 语言是目前比较流行、使用比较广泛的高级程序设计语言。C 语言的主要特色是兼顾了高级语言和汇编语言的特点，简洁，丰富，可移植性强。目前有许多系统软件和应用软件都是用 C 语言编写的。因此，C 语言受到了人们的极大推崇。

1.1 程序与程序设计语言

什么是程序及如何利用程序设计语言进行程序设计是初学者面临的第一个问题。本节将从程序、程序设计及程序设计语言的发展等方面加以阐释，以引导读者对程序设计的基本思想有一个全面的了解。

1.1.1 程序

在现实世界中，我们对“程序”一词其实并不陌生，而且也总是在不断地编写程序和执行程序。例如，我们要用全自动洗衣机洗衣服，应该如何去做呢？这就需要编写一个程序，洗衣服的程序如下所示：

(1)把要洗的衣服放入洗衣机。

(2)插好电源，打开水龙头。

(3)设置洗衣机的控制面板。

(4)放入洗涤剂。

(5)按下开始按钮。

(6)等待洗衣机清洗完毕。

清洗完毕后，蜂鸣器发出响声，这时可将衣服取出晾晒，这就是简单的程序形式。描述这种程序，也就是给出了一个包含其中各个基本步骤的序列。如果按顺序实施这些步骤的动作，其整体效果就是完成了该项工作。

在计算机中，程序是指导计算机执行某个功能或功能组合的一组指令。每一条指令都让计算机执行完成一个具体的操作，一个程序所规定的操作全部执行完毕后，就能产生计算结果。

计算机程序设计的过程一般由以下 4 个步骤组成：

(1)分析问题。在着手解决问题之前,应该通过分析充分理解问题,明确原始数据、解题要求、需要输出的数据及形式等。

(2)设计算法。算法就是一步步的解题过程。首先集中精力于算法的总体规划,然后逐层降低问题的抽象性,逐步充实细节,直到最终把抽象的问题具体化成可用程序语句表达的算法。这是一个自上而下、逐步细化的过程。

(3)编码。利用程序设计语言表示算法的过程称为编码。程序是一个用程序设计语言通过编码实现的算法。

(4)调试程序。程序调试包括编译和链接等操作。编译程序对程序员编写的源程序进行语法检查,程序员根据编译过程中的错误提示信息,查找并改正源程序的错误后再重新编译,直到没有语法错误为止,编译程序将源程序转换为目标程序。大多数程序设计语言往往还要使用链接程序把目标程序与系统提供的库文件进行链接以得到最终的可执行文件。

1.1.2 程序设计语言

人与人之间交换信息最重要的手段是语言,其包括口头语言、文字语言和图形语言等。但不同国家和不同民族的人有不同的语言,我们把这些语言统称为自然语言。目前,通用的计算机不能识别自然语言,只能识别特定的计算机语言。计算机语言,即程序设计语言,它是程序设计的工具,按其发展先后出现了机器语言、汇编语言和高级语言。

1. 机器语言

现代计算机以冯·诺依曼体系结构为主体,其内部采用二进制形式,即用 0 和 1 表示数据和指令,这种结构的计算机能够直接识别和处理由 0 和 1 编码组合而成的二进制代码,称为机器指令,一种计算机系统的全部机器指令的集合就是该计算机的机器语言。机器语言是一种面向机器的计算机语言,与计算机型号有关,每种计算机只能识别自己的机器语言。例如,001111 就是一条机器指令,通常表示向累加器送入一个数据。

用机器语言编写的计算机程序可以被计算机直接识别和处理,运算效率较高,但是缺点也是显而易见的,即这种程序只能在特定的计算机上使用,不具有可移植性。机器指令本身很难表达其含义,对于编程者来说记忆困难。每条机器指令的功能过于单一,编程效率低。

2. 汇编语言

机器语言的缺点制约了计算机的使用效率和使用范围,之后出现了汇编语言。汇编语言使用具有一定含义的符号表示机器指令,方便了编程人员的记忆和书写,提高了编程效率。例如,ADD AX,2 是一条汇编指令,通常表示将寄存器 AX 中存储的数据加上 2 再存入寄存器 AX 中。从指令本身可以直接看出这条指令能够完成加法操作。

与机器语言相比,汇编语言容易记忆、含义明确、便于编程人员使用。但是,使用汇编语言编写的程序不能直接被计算机识别与处理,必须经过处理将其转换成二进制机器指令执行,这种处理过程称为汇编,完成这种处理功能的程序称为汇编程序。汇编语言中的汇编指令与机器语言中的机器指令是一一对应的,汇编过程主要就是将助记符式的汇编指令转换为二进制机器指令的过程,因此汇编语言也是与机器相关的计算机语言。

3. 高级语言

高级语言的出现使计算机语言脱离了对计算机硬件的依赖,成为一种接近于人类自然语言的计算机语言,使高效编程与计算机的广泛应用成为可能,是计算机语言发展史上的一次飞跃。例如,“y=x+1;”就是一条高级语言的语句,表示将变量 x 的值加 1 再存入变量 y 中。

与汇编语言相比，高级语言具有以下优点：

(1)高级语言通常有一套特定的语法，这个语法与具体的计算机系统无关，用高级语言编写的程序可以独立于具体的计算机系统。也就是说，使用高级语言编写的程序，几乎不需要做任何修改就可以运行在支持该语言的计算机系统上，因此具有可移植性。

(2)高级语言更接近于人类常用的表述方式，语句的含义更加明显，容易学习与记忆。

(3)一条高级语言语句所完成的功能相当于多条机器指令可以完成的功能，因此编程效率较高。

自从高级语言问世以来，曾得到广泛应用的高级语言有 BASIC、FORTRAN、COBOL、PASCAL、C、Ada、LISP 等，这些语言都有各自的特点和适用领域。例如，FORTRAN 适用于数值计算、COBOL 适用于商业管理、C 适用于编写系统软件。

目前，面向对象编程语言已经成为继上述高级语言之后日益成熟并得到广泛应用的计算机语言，如 Smalltalk、C++、Java 等，而且将可视化、事件驱动等新技术与计算机语言结合在一起构建的各种集成开发环境，已成为应用软件开发的重要工具，但是 C 语言的基础性作用不容忽视。

1.2 C 语言的发展及主要特点

C 语言因其功能强大和诸多优点，自诞生以来便逐渐为人们所认识。到了 20 世纪 80 年代，C 语言很快在各大、中、小和微型计算机上得到了广泛应用，成为当时最优秀的程序设计语言之一。

1.2.1 C 语言的发展历史

1978 年，美国电话电报公司(AT&T)的贝尔实验室正式发布了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的 *The C Programming Language* 一书，通常简称为 K&R，也称为 K&R 标准。但是，在 K&R 中并没有定义一个完整的标准 C 语言。

1983 年，美国国家标准协会(American National Standards Institute，ANSI)认识到标准化将有助于 C 语言在商业化编程中的普及，因此成立了一个委员会为 C 语言及其运行制定标准，即 ANSI C 标准。

1989 年，国际标准化组织(ISO)采纳了 ANSI C 标准，称为 ANSI/ISO standard C(ANSI x3. 159—1989)，通常称为 ANSI C89 标准。之后，C 语言标准在很长一段时间内都保持不变，在 20 世纪 90 年代才经历了改进，这就是 ISO/IEC 9899:1999，简称 C99，它被 ANSI 于 2000 年 3 月采用。

2011 年，ISO 发布了 C 语言的新标准 C11，官方称为 ISO/IEC 9899:2011。新的标准提高了对 C++ 的兼容性，并增加了泛型宏、多线程、匿名结构等新特性。

在 C 语言的发展过程中，出现了多种版本的 C 语言。例如，Microsoft 公司开发的Microsoft C 或称 MS C，Borland 公司开发的 Turbo C 和 Borland C，AT&T 公司开发的 AT&T C 等，这些 C 语言版本不仅遵守了 ANSI C 标准，而且在此基础上各自进行了一些扩充，使之更加方便易用。

1.2.2 C 语言的主要特点

在众多高级程序设计语言中，C 语言之所以能够存在和发展，并具有旺盛的生命力，主要源于其自身的优良特性。这些特点可概括为以下几点：

(1)C 语言的标识符要求区分大小写。C 语言以英文小写字母为基础，符合人们日常阅读和书写的习惯。同一个单词的大小写代表不同含义的标识符。

(2)语言简洁、紧凑，使用方便、灵活。C 语言一共有 32 个关键字、9 种控制语句，程序书写形式自由，压缩了一切不必要的成分。

(3)模块化程序设计。C 语言由函数组成，因此程序功能结构比较清楚，而且每个函数都是独立

的,可以单独进行编译。

(4)运算符丰富。在 C 语言中,括号、赋值、强制类型转换等都作为运算符处理,从而使 C 语言的运算类型极其丰富,表达式类型多样化,灵活使用各种运算符可以实现其他高级语言难以实现的运算。

(5)数据结构丰富。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、共用体类型等,能用来实现各种复杂的数据结构,如链表、树、栈等的运算。

(6)具有结构化的控制语句。C 语言具有典型的结构化控制语句。例如,选择语句(if 语句和 switch 语句)、循环语句(while 语句、do-while 语句和 for 语句)等均符合现代编程风格的要求。

(7)语法限制不太严格,程序设计自由度大。例如,对数组下标越界不做检查,由程序编写者自己保证程序的正确。对变量的类型使用比较灵活。例如,整型变量与字符型数据及逻辑型数据可以通用。

(8)C 语言允许直接访问物理地址,能够进行位操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此,C 语言既具有高级语言的功能,又具有低级语言的许多功能,可以用来编写系统软件,因此有人把 C 语言也称为中级语言。

(9)生成的目标代码质量高,程序执行效率高。

(10)可移植性好。用 C 语言编写的程序基本上不用进行修改就能用于各种操作系统。

1.3 初识 C 语言程序

C 语言作为一种高级程序设计语言,其语法与自然语言具有一定的相似性,因此在学习 C 语言之前,有必要对 C 语言程序的基本结构有一个初步的感性认识。下面从最简单的例子着手,来分析 C 语言程序的构成及运行情况。

【例 1-1】 编写一个简单的 C 语言程序,用于输出指定信息。

```
// FileName: chap1_1.c
#include <stdio.h>
int main( )
{
    printf("This is the first C program.\n");        //输出语句
    return 0;
}
```

程序运行结果如下:

```
This is the first C program.
```

【例 1-1】是一个非常简单的程序,但它体现了 C 语言的几个重要组成部分。下面分别对该程序中的各行加以分析。

程序中第 1 行以//开始的一段文字称为注释,注释文字可以由任意字符组成。注释不参与程序的运行,主要用于对程序的某些关键部分进行说明,其目的是提高程序的可读性。因此,在程序的适当位置添加必要的注释是一种良好的编程习惯,注释可以出现在程序中的任意地方。

第 3~7 行是该程序的主要组成部分,在 C 语言中被称为主函数,函数名为 main。可见,C 语言程序是由函数构成的。任何 C 语言程序有且仅有一个主函数,主函数可以出现在程序的任意位置,C 语言程序就从这个主函数开始执行。

第4～7行是函数main( )的函数体，由花括号{ }括起来。左花括号是函数体的开始，右花括号是函数体的结束。这一对花括号及其中的程序也被称为程序块。

在函数main( )中，第5行和第6行是完成函数功能的主要部分，在C语言中被称为语句。每一条语句都以“;”作为结束标记。可见，C函数是由语句构成的。

程序第5行语句中的printf是另一个函数的名字，功能是输出其后圆括号中的字符串。为了编程方便，系统提供了许多标准函数供用户使用，这样的函数被称为库函数。使用这些库函数之前，需要在程序的开始包含该库函数所在的头文件，即该程序中的第2行。

在第6行的语句中可以看到一个英文单词return，它在C语言中有特殊的作用，即结束本函数的执行，返回函数的调用处。这种规定了特殊作用的单词称为关键字，关键字是C语言语句的一种重要组成要素，每个关键字都表示某种特定的意义。

另外，C语言不关心程序在文本行的开始位置，可以在任意位置输入程序。因此，编程人员在输入源程序代码时，可以对源程序进行排版，使用Tab键缩进某些行是一种较好的排版方式，这样写出的程序层次分明，更易于阅读和理解。

1.4 C语言程序的调试

利用C语言编写程序的最终目的是高效地解决现实世界各领域中的实际问题，对实际问题进行分析，以C语言构建程序的思想为指引设计解决问题的方案，是构建C语言程序的第一步，通常被称为程序设计。在此基础上，按照C语言的规则编写出C语言程序并将其存储在计算机中，运行后产生正确的结果，是构建C语言程序的第二步，这通常被称为程序生成。经过这两步的工作，达到了以计算机为工具解决实际问题的目的。在程序生成过程中，编程人员还需要不断检查程序中的语法错误，反复修改，直至得到正确的结果。

1.4.1 C语言程序的开发过程

从确定C语言程序算法开始编写代码到上机运行得到结果，C语言程序的开发过程一般包括以下几个步骤：

1. 程序编辑

把编写好的程序输入计算机，以文件的形式存储在磁盘中的过程，被称为程序编辑。能够完成这项工作的软件被称为编辑软件(也被称为编辑器)。在编辑方式下建立起来的程序文件被称为源程序文件，简称源文件，源文件中的程序被称为源程序。绝大多数C语言编译器将文件名结尾为.c的文件看作C语言程序的源文件(Visual C++ 2010环境下生成的C语言程序源文件扩展名为.cpp)。例如，可以将【例1-1】中的C语言源程序输入计算机并保存到一个名为chap1_1.c的源文件中。

2. 程序编译

程序编译是指把编辑好的源程序翻译成目标代码的过程。能够完成这项工作的软件被称为编译软件(也被称为编译器)。目标代码是指计算机能识别的二进制代码，存放目标代码的文件被称为目标文件。通常情况下，目标文件名与源文件名相同，文件扩展名为.obj。但不同的操作系统与编译软件也可能采用不同形式的后缀。

在程序编译过程中，对源程序中的每一条语句，编译器都要进行语法检查，当发现错误时，会在屏幕上显示错误位置和错误类型的信息。用户看到这类提示信息后，要再次调用编辑器对源程序中的错误进行修改，然后再次编译，直到排除所有的语法和语义错误。正确的源程序经过编译后在磁盘上生成目标文件。例如，对源文件chap1_1.c编译完成后，系统会自动生成一个名为chap1_1.obj的目

标文件,存放在源文件所在的存储路径下。

3. 程序链接

编译后产生的目标文件不能在机器上直接运行。程序中会用到库函数或者其他函数,它们都是可重定位的程序模块,需要把它们连成一个统一的整体,这个过程被称为程序链接。目标代码经过链接后,生成可以运行的可执行程序,存储可执行程序的文件被称为可执行文件,通常存放在目标文件所在的存储路径下。通常情况下,可执行文件名也与源文件名相同,文件扩展名为. exe。例如,对目标文件 chap1_1. obj 完成链接后,系统会自动生成一个名为 chap1_1. exe 的可执行文件。

4. 程序运行

生成可执行文件后,程序就可以在操作系统控制下运行了。若执行程序后达到了预期目的,则 C 语言程序的开发工作到此完成。否则,要进一步对程序进行调试,重复编辑、编译、链接及运行的过程,直到取得预期结果。调试是指发现程序中的错误并改正错误的过程,是任何程序开发都需要经历的一个非常重要的过程,需要编程人员发挥聪明才智,根据错误的表现分析错误的原因并找到错误的位置,然后进行正确的修改。

图 1-1 所示为开发一个 C 语言程序的基本过程。

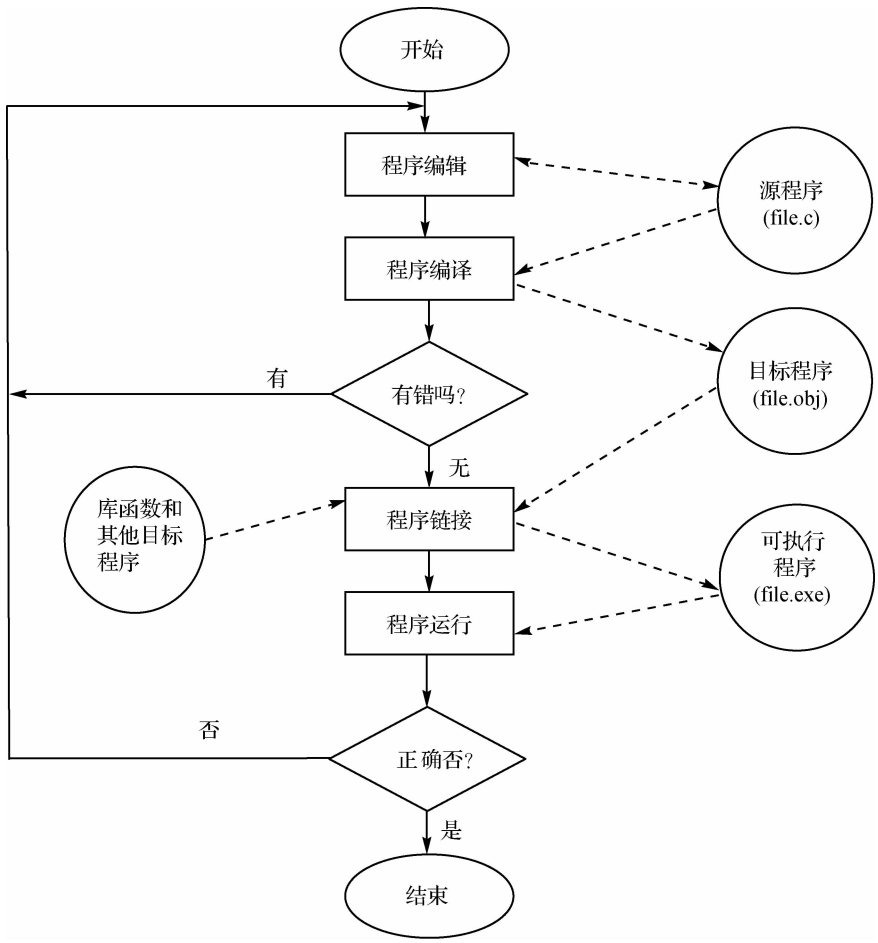

图 1-1　C 语言程序开发流程图

1.4.2 C语言程序的上机步骤

目前，有多种可用的C程序开发环境，如Turbo C、Borland C、Visual C++ 6.0、Visual C++ 2010、DEV-C等。这些开发工具已将程序的编辑、编译、链接和运行的过程集成在一起，由单个应用程序来完成上述四项工作，并提供了多种帮助用户书写和修改程序的手段，这样的应用程序被称为集成开发环境(integrated development environment，IDE)。IDE通常基于窗口环境，在窗口中完成对程序的编辑、编译、链接、运行和调试的全部工作，使用IDE工具可以大大简化应用程序的开发过程。

本书以Visual C++ 2010(以下简称为VC++ 2010)作为C语言程序的集成开发环境。VC++ 2010是Microsoft公司的Visual Studio 2010开发工具箱中的一个C++程序开发包，从早期版本发展到现在，已经有了很大的变化，在界面、功能、库支持等方面都有许多的增强。

VC++ 2010从编辑到运行一个C语言程序的步骤如下：

(1)启动VC++ 2010集成开发环境。

(2)新建项目，定位合适的路径保存。

(3)添加源文件，源文件的扩展名为.c或.cpp。

(4)编辑(或修改)源文件。

(5)生成解决方案。如果生成成功，则会生成扩展名为.exe的可执行程序；否则，根据显示的错误信息进行修改，再生成解决方案，直至成功。

(6)运行可执行程序。通过观察程序运行结果，验证程序的正确性，如果结果不正确，回到步骤(4)，重复步骤(4)和步骤(5)，直至程序结果正确。

(7)退出VC++ 2010集成开发环境，结束本次程序运行。

下面通过实例来描述从编辑到运行一个C语言程序的过程。

1. 新建项目

(1)在Windows操作系统的程序列表中找到VC++ 2010图标，启动VC++ 2010。

(2)执行“文件(File)”→“新建(New)”→“项目(Project)”命令，在弹出的“新建项目”对话框中选择“Win32控制台应用程序(Win32 Console Application)”，如图1-2所示。

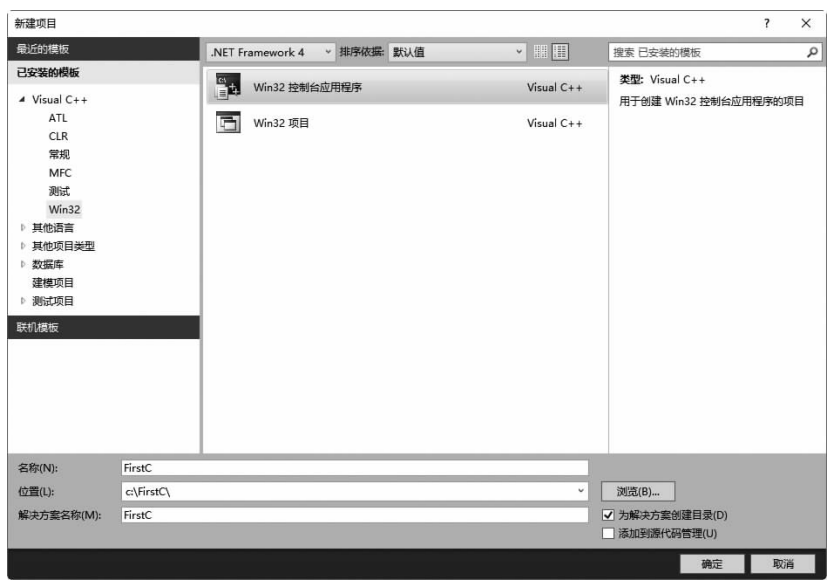

图1-2 “新建项目”对话框

(3)在“位置(Location)”文本框中输入存储路径,在“名称(Name)”文本框中输入项目名称,然后单击“确定(OK)”按钮。

(4)在 Win32 应用程序向导的“概述(Overview)”界面直接单击“下一步(Next)”按钮。

(5)在 Win32 应用程序向导的“应用程序设置(Application Settings)”界面选中“空项目(Empty Project)”复选框,如图 1-3 所示,然后单击“完成(Finish)”按钮。

图 1-3 “应用程序设置”界面

2. 编辑源程序

(1)执行“视图(View)”→“解决方案资源管理器(Solution Explorer)”命令,在解决方案资源管理器中,右击“源文件(Source Files)”,在弹出的快捷菜单中选择“添加(Add)”→“新建项(New Item)”命令,弹出“添加新项”对话框,选择“C++ 文件(C++ File)”选项,然后在“名称(Name)”文本框中输入源文件名称,如图 1-4 所示,最后单击“添加(Add)”按钮。

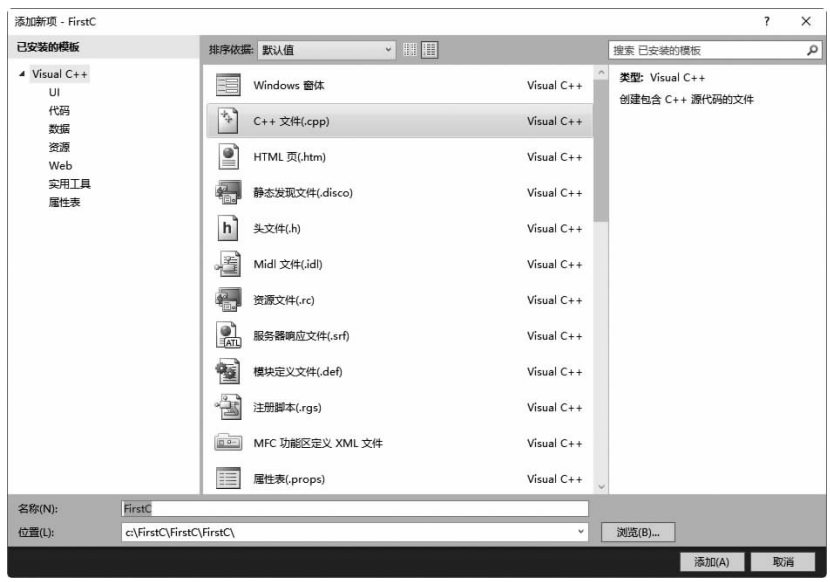

图 1-4 新建源文件

(2)在编辑区中输入源程序代码,如图 1-5 所示。

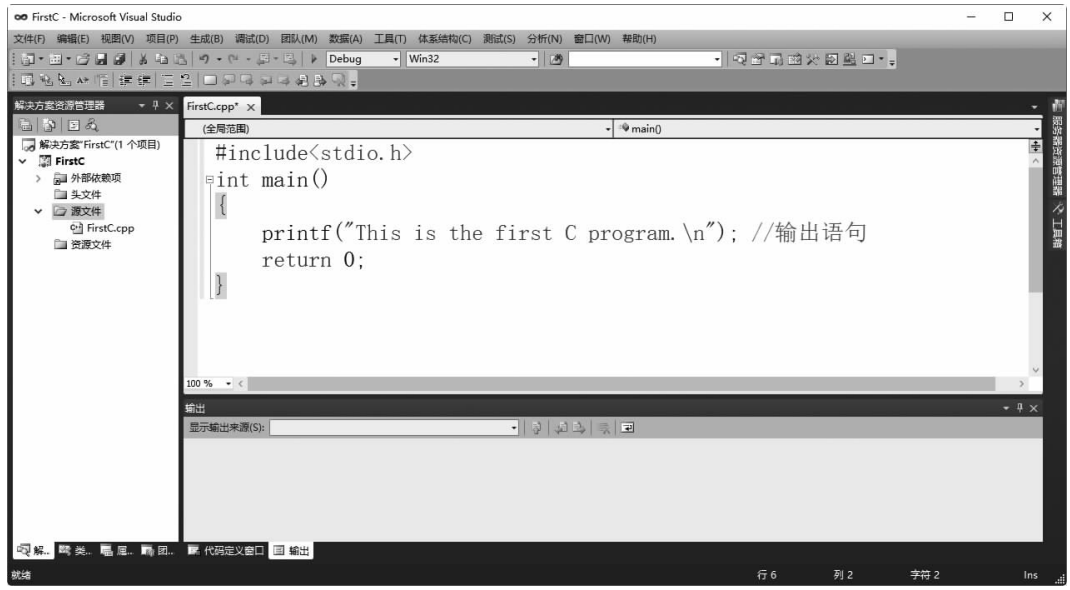

图 1-5 编辑源程序

3. 生成解决方案

(1)执行“生成(Build)”→“生成解决方案(Build Solution)”命令,对源程序进行编译链接。

(2)如果解决方案生成成功,将在如图 1-6 所示的窗口中显示“生成:成功 1 个,失败 0 个,最新 0 个,跳过 0 个”,表示没有任何错误。

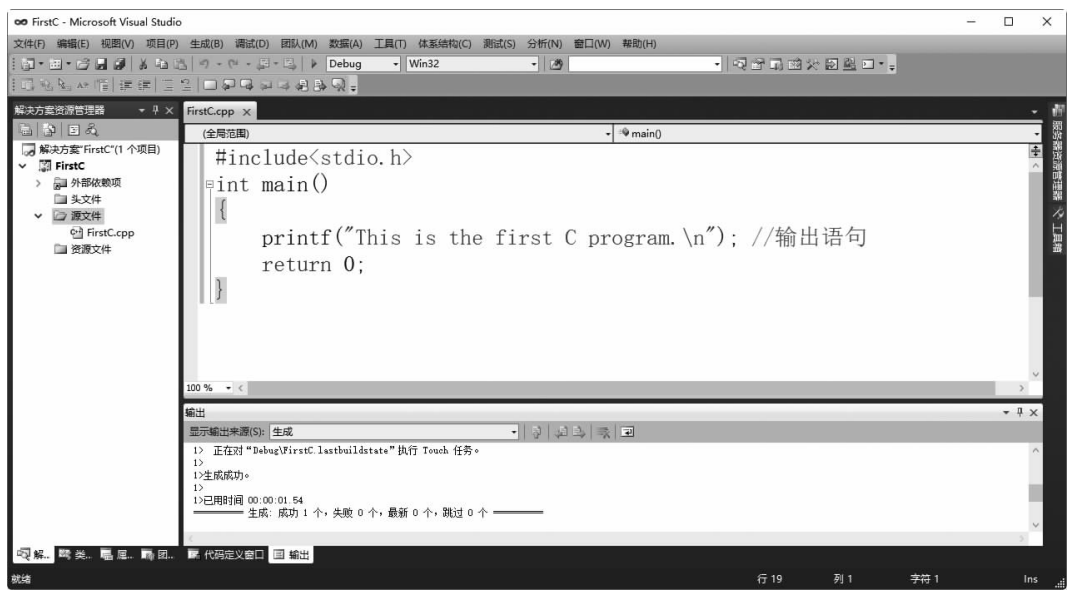

图 1-6 生成解决方案

4. 运行程序

执行“调试(Debug)”→“开始执行(不调试)(Start Without Debugging)”命令运行程序,可执行程序运行后,将显示为 DOS 控制台状态,如图 1-7 所示。按任意键返回到 VC++ 2010 环境。

图 1-7　运行程序

1.5　等级考试内容与真题解析

1.5.1　考试内容

全国计算机等级考试二级 C 语言程序设计考试大纲(2018 年版)中针对本章的考试内容要求如下：

(1)程序的构成，main()函数和其他函数。

(2)头文件，数据说明，函数的开始和结束标志以及程序中的注释。

(3)源程序的书写格式。

(4)C 语言的风格。

1.5.2　真题解析

(1)以下叙述中正确的是(　　)。

A. C 程序的基本组成单位是语句　　B. C 程序中的每一行只能写一条语句

C. 简单 C 语句必须以分号结束　　D. C 语句必须在一行内写完

解析　C 程序的基本组成单位是函数，C 程序书写格式自由，一行可以写多条语句，一条语句也可以写在多行，简单 C 语句必须以分号作为语句的结束标识。故选 C。

(2)计算机能直接执行的程序是(　　)。

A. 源程序　　B. 目标程序

C. 汇编程序　　D. 可执行程序

解析　用高级语言和汇编语言编写的程序不能够被计算机直接执行，必须被翻译成机器语言程序，计算机才能直接执行，可执行程序是机器语言程序。故选 D。

(3)以下叙述正确的是(　　)。

A. C 程序中的注释只能出现在程序的开始位置和语句的后面

B. C 程序书写格式严格，要求一行只能写一条语句

C. C 程序书写格式自由，一条语句可以写在多行上

D. 用 C 语言编写的程序只能放在一个程序文件中

解析　C 程序中的注释可以放在程序中的任意位置，用 C 语言编写的程序可以放在多个文件中。故选 C。

(4)下列叙述中,不符合良好程序设计风格的是(　　)。

A. 程序的效率第一,清晰第二

B. 程序的可读性好

C. 程序中有必要的注释

D. 输入数据前要有提示信息

解析 良好的程序设计要求可读性好,程序中有必要的注释,输入数据前要有提示信息。故选A。

(5)以下有关C语言的叙述正确的是(　　)。

A. C语言程序将从源程序中第一个函数开始执行

B. 可以在程序中由用户指定任意一个函数作为主函数,程序将从此开始执行

C. C语言规定必须用main作为主函数名,程序将从此开始执行,在此结束

D. main可作为用户标识符,用以命名任意一个函数作为主函数

解析 C语言规定必须用main作为主函数,不管main函数放在什么位置,程序总是从main函数开始执行,最后在main函数中结束。故选C。

1.6 习题

(1)什么是程序?

(2)计算机语言经历了哪几个阶段?

(3)程序的翻译方式有哪几种?

(4)编写一个简单的C语言程序,使得在屏幕上显示下列信息:

```
**************************************************
              C is very fun.
**************************************************
```

(5)编写一个简单的C语言程序,使得在屏幕上显示下列信息:

```
    *
   ***
  *****
 *******
```

第2章

C语言的基本知识

作为一种程序设计语言，C语言有一套严格的字符集和语法规则，程序设计人员根据实际问题的需要，使用这些字符和语法规则编写C语言程序。本章主要介绍C语言的构成元素、C语言的基本数据类型、常量、变量、运算符及表达式等相关概念，并从实际编程的角度介绍其具体用法。

2.1 标识符和关键字

在人们学习汉语、英语等语言时，已经了解到一门语言大体都是由本语言的符号、字、词、句、段落和文章等组成的。C语言作为一门程序设计语言，其构成元素与汉语、英语等人类自然语言有些类似，见表2-1。

表2-1 自然语言与C语言组成要素的对比

自然语言	字	词		句	段	章
		单词	短语			
C语言	字符	标识符	表达式	语句	函数	程序

字符是C语言最基本的语言要素，采用一种编码形式描述，即美国国家标准信息交换码(American standard code for information interchange，ASCII)，将所有的字符组织在一起形成C语言的字符集。将字符集中的字符按照一定的规则进行组织，构成了C语言中的关键字或标识符。将它们按照C语言规定的语法规则进行组织，构成了C语言中的各种语句。根据要完成的特定功能将某些语句按照一定的规则组织在一起，构成了C语言的函数。多个函数组合在一起构成了C语言程序，该程序能够完成指定的功能。因此，按照“字—词—句—段—章”的自然语言的学习顺序来学习C语言是一种非常有效的学习方法。

1. 字符集

字符是组成语言的最基本元素，国际上使用最广泛的计算机字符编码是ASCII码，标准ASCII码字符集包括128个字符，主要由字母字符、数字字符、空格符、特殊字符和其他字符组成。

(1)字母字符。字母字符包括大写字母A～Z及小写字母a～z共52个字符。在ASCII码表中，字母编码的排列顺序符合通常的自然语言顺序，而且对应的大、小写字母的ASCII码值相差32。例如，大写字母A的ASCII码值为65，大写字母B的ASCII码值为66，以此类推。与之相对应，小写字母a的ASCII码值为65＋32＝97，小写字母b的ASCII码值为66＋32＝98。

(2)数字字符。数字字符包括0～9共10个字符。在ASCII码表中,数字编码的排列顺序也满足通常的顺序,即0～9,并且用ASCII码的十六进制值表示时,个位码正好与其对应的数字相同。例如,数字0的ASCII码值为48,用十六进制表示为30;数字9的ASCII码值为57,用十六进制表示为39。

(3)空格符。空格符只在字符常量和字符串常量中起作用。在其他地方出现时,只起间隔作用,因此在程序中使用一个或多个空格符对程序的编译不产生影响,但在程序中恰当地使用空格符能增加程序的清晰性和可读性。

(4)特殊字符。特殊字符是不可显示、不可打印的字符,用于计算机设备的操作控制及在数据通信时进行传输控制,因此也称为控制符。例如,FP表示换页,用于在打印输出时进行换页控制。

(5)其他字符。其他字符包括图形符、标点符和运算符等。例如,$的ASCII码值为36,运算符+的ASCII码值为43。

2. 标识符

所谓标识符就是用来标识在C语言程序中出现的符号常量、变量、数据类型、函数和语句的字符序列,C语言中的标识符由字符组成,满足一定的构成规则。C语言规定,标识符由字母、数字和下划线组成,且第一个字符不能使用数字字符。例如,a、_3x、BOOK1、sum5、student_1等都是合法的标识符,而-3x、bowy-1、3Student、Sum@Mul、a>b等都是不合法的,不能用作C语言的标识符。

在符合标识符命名规则的前提下,在程序中使用含义清晰的标识符能够提高程序的可读性和可理解性,从而为程序的编写和维护提供方便,因此,建议程序编写者在命名标识符时尽量做到见名知意,尽量避免使用没有实际含义的标识符。例如,把存储面积值的标识符命名为area,把表示累加和的标识符命名为sum,而不使用value1、abc等这样的标识符。

用户在程序中自定义标识符时必须注意以下几点:

(1)关键字和特定标识符不能作为用户自定义标识符。

(2)标准C不限制标识符的长度,但有些C语言编译系统限制标识符的长度,同时标识符的长度也受到具体处理器的限制。

(3)在标识符中区分大小写字母。例如,Book和book是两个完全不同的标识符。

(4)避免使用易混淆的字符,如整数1和小写字母l,数字0和小写字母o,数字2和小写字母z等,尤其对于初学者来说,这种错误在调试过程中很难被发现。

(5)用户定义标识符不要与C语言的库函数同名。如果出现这种情况,库函数就将失去原有含义。

3. 关键字

关键字是系统定义的、具有特定含义、专门用于特定用途的C语言标识符,也被称为保留字。关键字一般为小写字母,在使用时必须遵循一定的语法规则,如果随意使用关键字,可能会出现意想不到的错误,有时程序虽然编译通过,但运行结果却不正确,并且很难检查出来。标准C语言中共有32个关键字,见表2-2。

表 2-2　C 语言的关键字

关 键 字	含　　义	类　　型
int	整型	数据类型
short	短整型	
long	长整型	
float	单精度浮点型	
double	双精度浮点型	
char	字符型	
void	无值型	
unsigned	无符号型	
signed	有符号型	
const	常量	
struct	结构体型	
union	联合型	
enum	枚举型	
volatile	易变型	
sizeof	求字节数	运算符
if	条件语句	流程控制
else	与 if 配合使用	
switch	开关语句	
case	与 switch 配合使用	
default	与 switch 配合使用	
for	循环语句	
while	循环语句	
do	循环语句	
break	间断语句	
continue	接续语句	
return	返回语句	
goto	跳转语句	
auto	自动类型	存储类型
extern	外部类型	
static	静态类型	
register	寄存器类型	
typedef	用户自定义类型命名	

另外,C99 标准中新增了一些关键字,如_Bool、_Complex、_Imaginary、inline 和restrict。

2.2 数据类型

所谓数据类型是指被定义变量的性质，是按表示形式、占据存储空间的多少和构造特点来划分的。在 C 语言中，数据类型可分为基本数据类型、构造数据类型、指针数据类型和空类型，如图 2-1 所示。

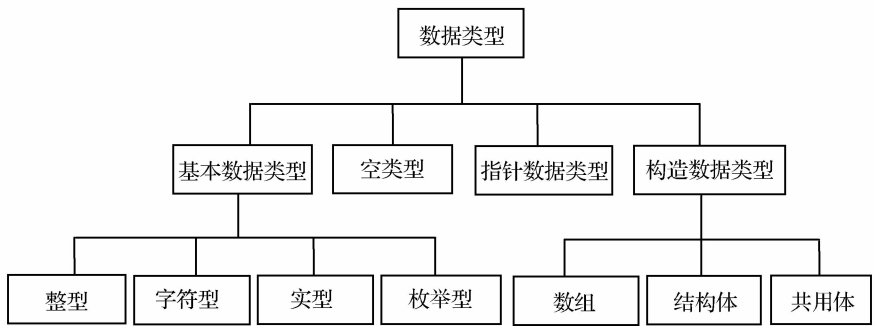

图 2-1 C 语言的数据类型

C 语言程序中用到的数据在内存中都要根据其对应的数据类型分配一定大小的内存空间，分配的内存空间的大小与具体的硬件和编译软件有关，本书在不加说明的情况下均以 16 位机 VC 编译器为例。本章主要介绍基本数据类型，其他数据类型将在后续章节中学习。ANSI C 标准规定了整型、实型和字符型的最小长度和数值范围，见表 2-3。

表 2-3 基本类型的最小长度和数值范围

类型名称	中文名称	字节数/byte	位数/bit	数值范围	备　注
char	字符型	1	8	−128～127	$-2^7 \sim (2^7-1)$
int	整型	2	16	−32 768～32 767	$-2^{15} \sim (2^{15}-1)$
float	单精度实型	4	32	$-3.4\times10^{-38} \sim 3.4\times10^{38}$	6～7 位有效数字
double	双精度实型	8	64	$-1.7\times10^{-308} \sim 1.7\times10^{308}$	15～16 位有效数字

ANSI C 标准规定，简单类型的前面还可以加上修饰符，从而使简单类型的语义更加丰富，方便 C 语言编程人员选用恰当的数据类型。这样的修饰符共有 4 种，即 signed(有符号)、unsigned(无符号)、long(长型)和 short(短型)。组合后形成的类型见表 2-4。

表 2-4 ANSI C 标准中基本类型的最小长度和数值范围

数据类型	中文名称	字节数/byte	位数/bit	数值范围	备　注
char	字符型	1	8	−128～127	
unsigned char	无符号字符型	1	8	0～255	
signed char	有符号字符型	1	8	同 char	
int	整型	2	16	−32 768～32 767	

续表

数据类型	中文名称	字节数/byte	位数/bit	数值范围	备　注
unsigned int	无符号整型	2	16	0～65 535	可省略整型说明符 int
signed int	有符号整型	2	16	同 int	
short int	短整型	2	16	−32 768～32 767	
unsigned short int	无符号短整型	2	16	0～65 535	
signed short int	有符号短整型	2	16	同 short int	
long int	长整型	4	32	-2^{31}～$2^{31}-1$	
unsigned long int	无符号长整型	4	32	0～$2^{32}-1$	
signed long int	有符号长整型	4	32	同 long int	
float	单精度实型	4	32	-3.4×10^{-38}～3.4×10^{38}	6～7 位有效数字
double	双精度实型	8	64	-1.7×10^{-308}～1.7×10^{308}	15～16 位有效数字

另外,C99 标准在 C89 标准基础上进行了一些修改,增加了_Bool、_Complex 和_Imaginary 三种基本类型,同时还增加了修饰符 long long,即出现了 long long int、unsigned long long int 等数据类型,并允许以 LL 或 ll 为后缀来表示 long long 型。

2.3　常量与变量

对于基本数据类型量,按其取值是否可改变又分为常量和变量两种。在程序中,常量是可以不经说明而直接引用的,而变量则必须先定义,后使用。

2.3.1　常量与变量的概念

1. 常量

在程序运行过程中值保持不变的量被称为常量。常量可以分为符号常量和直接常量两种。符号常量是指用标识符预定义的常量,从字面上不能直接看出其类型和值。直接常量又称字面常量,包括整型常量、浮点型常量、字符常量和字符串常量。

在 C 语言中,直接常量是直接以自身的存在形式体现值和类型的。例如,123、−5 是整型常量,1.5、−1.2E−2 是实型常量,'x' 是字符常量,"first"是字符串常量。

在 C 语言中,符号常量是采用宏定义命令定义的常量,定义形式如下:

```
#define 符号常量名　常量
```

其中,符号常量名应遵循标识符的命名规则,#define 是宏定义的专用定义符,将在后面章节中进行详细讲解。例如,#define PI 3.14159 表示定义 PI 为一个符号常量,C 语言编译系统在处理程序时会将程序中的全部 PI 均用 3.14159 代替。

2. 变量

在程序运行过程中值可以被改变的量被称为变量。变量在内存中根据其数据类型占据大小不同的存储单元,用来存入可能变化的值。

变量包括变量类型、变量名和变量值3个基本概念。变量类型表明变量用来存放什么类型的数据，变量名用来区分并引用不同的变量，在变量的存储单元中存放的数据被称为变量值。

(1)变量的定义。C语言中的变量遵循“先定义，后使用”的原则，就是必须先对将要使用的变量进行定义，说明变量的数据类型，然后才能使用该变量。这样做的目的如下：

①变量定义是为变量指定数据类型。确定变量的数据类型后，在编译时就能在内存中分配相应的存储单元。

②能保证变量名的正确使用。

③便于在编译时根据变量的数据类型检查该变量所做的运算是否合法。例如，只有整型变量之间才能进行求余运算。

变量定义的一般形式如下：

```
类型说明符  变量名标识符,变量名标识符,…;
```

其中，类型说明符用于确定变量的数据类型，变量名标识符必须符合用户自定义标识符的命名规则。举例如下：

```
int i,j,k;          //定义 i,j,k 为整型变量
float f1,f2;        //定义 f1,f2 为单精度实型变量
char c1,c2;         //定义 c1,c2 为字符型变量
```

在定义变量时应该注意以下几点：

①变量的定义必须在变量使用之前进行，一般放在函数体开头的声明部分。

②允许同时定义同一数据类型的多个变量，多个变量之间用“,”分隔。

③最后一个变量名后必须以“;”结束。

④类型说明符与变量名之间至少要有一个空格。

(2)对变量赋初值。C语言中允许在定义变量的同时对变量赋初值，这也被称为变量的初始化。

举例如下：

```
int a = 2;
char c = 'x';
float x = 1.2, y = 2.4;
```

(3)对变量的基本操作。变量可以被看成存储数据的容器。有两种对变量的基本操作：一是向变量中存入数据，这种操作被称为赋值；二是取得变量当前的值，以便在程序运行时使用，这种操作被称为取值。

举例如下：

```
a = 2;
b = a + 3;
```

第一条语句是将2赋值给变量a，即变量a对应的存储单元中存放2。第二条语句先将变量a的当前值2取出，加上3后得到5的结果再赋值给变量b。

2.3.2 整型常量与变量

整型数据没有小数部分，根据占有存储空间的长度，分为基本整型、短整型和长整型3种，类型说明符分别为int、short int(或short)及long int(或long)。根据存储单元中是否有符号位，整型数据又

可分为有符号类型和无符号类型。对于有符号类型,在存放整数的存储单元中,最左面的一位表示符号,该位为 0 时,表示数值为正;该位为 1 时,表示数值为负。

整型数据在内存中是以二进制方式存放的,最高位为符号位,并以补码表示。将一个十进制整数转化为补码表示的方法如下:

(1)对于正数,其补码表示与原码相同。

(2)对于负数,其补码表示为反码加 1(负数的反码为其绝对值的所有位取反)。

例如,求－10 的补码的方法如下:

①取－10 的绝对值 10。

②10 的二进制形式为 0000000000001010(一个整数占 16 位)。

③对 0000000000001010 取反得到 1111111111110101。

④再加 1 得 1111111111110110,如图 2-2 所示。

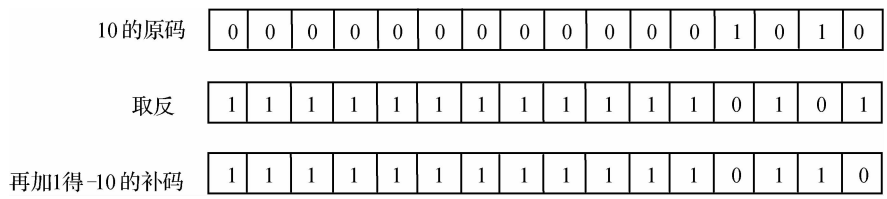

图 2-2　求－10 的补码

1. 整型常量

在 C 语言中,整型常量可以表示为十进制、八进制和十六进制 3 种。

(1)十进制整型常量。十进制整型常量的形式为 d。其中,d 可以是 0～9 的一个或多个十进制数,首位不能为 0。进位规则为逢十进一。例如,236、－578、65 535、1 627 均是合法的十进制整型常量,而 025、23A 是非法的十进制整型常量,因为 025 首位为 0,23A 中含有非十进制数码 A。

(2)八进制整型常量。八进制整型常量的形式为 0d。其中,d 可以是 0～7 的一个或多个八进制数,起始 0 是必需的引导符。进位规则为逢八进一。例如,016(十进制的 14)、0111(十进制的 73)均是合法的八进制整型常量,而 256、03A2 是非法的八进制整型常量,因为 256 没有前缀 0,03A2 中含有非八进制数码 A。

(3)十六进制整型常量。十六进制整型常量的形式为 0xd。其中,d 可以是一个或多个十六进制数(0～9 的数字,或 a～f 的字母)。引导符 0 是必需的,字母 X 可以用大写或小写,进位规则为逢十六进一。例如,0X2B(十进制为 43)、0XA0(十进制为 160)、0XFFFF(十进制为 65 535)均是合法的十六进制整型常量,而 5A、0X3H 是非法的十六进制常量,因为 5A 没有前缀 0X,0X3H 中含有非十六进制数码 H。

需要强调的是,十进制、八进制和十六进制只是整数的三种不同的表示形式,在计算机内部都将转换成相应的二进制数存储。因此,同一个整数可以有 3 种不同的表示方法。下面的例子分别给出了整数 10 的十进制、八进制和十六进制表示。

```
10     // 十进制整数 10,在内存中对应二进制数 0000000000001010
012    // 八进制整数 12,在内存中对应二进制数 0000000000001010
0xa    // 十六进制数 a,在内存中对应二进制数 0000000000001010
```

默认情况下,在－32 768～32 767 范围内的整型常数的数据类型是 int 型,超过此范围而在－2 147 483 648～2 147 483 647 范围内的整型常数的数据类型是 long 型。也可以通过在整型常数

后面加上字母后缀来强制指定其数据类型，C语言规定的字母后缀的具体含义如下：

- 后缀 l 或 L 表示 long 型常数。例如，－12l、01235456720L。
- 后缀 u 或 U 表示 unsigned 型常数。例如，12u、034u、0x2fdU。
- 后缀 lu 或 LU 表示 unsigned long 型常数。例如，123246875LU。

2. 整型变量

整型变量可以分为以下几类：

(1)基本整型。类型说明符为 int，在内存中占 2 字节，其取值范围为－32 768～32 767。

(2)短整型。类型说明符为 short int 或 short，在内存中所占字节数和取值范围均与基本整型相同。

(3)长整型。类型说明符为 long int 或 long，在内存中占 4 字节，其取值范围为－2 147 483 648～2 147 483 647。

(4)无符号型。类型说明符为 unsigned，此类型的整数没有负数，可分为以下 3 种：

①无符号基本整型。类型说明符为 unsigned int 或 unsigned，在内存中占 2 字节，其取值范围为 0～65 535。

②无符号短整型。类型说明符为 unsigned short int 或 unsigned short，在内存中所占字节数和取值范围均与无符号基本整型相同。

③无符号长整型。类型说明符为 unsigned long int 或 unsigned long，在内存中占 4 字节，其取值范围为 0～4 294 967 295。

整型变量在使用过程中要先定义后使用。

【例 2-1】 整型变量的定义与使用。

```
//FileName: chap2_1.c
#include <stdio.h>
int main( )
{
    int a,b,c,d;                    //定义 a、b、c、d 为基本整型变量
    unsigned u;                     //定义 u 为无符号基本整型变量
    a = 10; b = -20; u = 5;
    c = a + u;
    d = b - u;
    printf("c = %d,d = %d\n",c, d);
    return 0;
}
```

程序运行结果如下：

```
c = 15,d = -25
```

2.3.3 实型常量与变量

实型数据又称为浮点型数据，是带小数部分的数据。根据能够表示的大小和精度，浮点型数据分为单精度、双精度两类，类型说明符分别为 float 和 double。浮点型数据在内存中以指数形式存储。C语言将一个浮点型数据分成小数和指数两个部分存储。例如，实数 3.141 59 在内存中的存放形式

如图 2-3 所示。

图 2-3　实数 3.141 59 在内存中的存放形式

图 2-3 是用十进制数来表示的,在计算机中实际上是用二进制数来表示符号和小数部分,用 2 的幂次来表示指数部分。分配给实型数的若干字节中,到底使用多少位来表示小数部分及指数部分,ANSI C 未做具体规定,由 C 编译系统自行决定。

1. 实型常量

在 C 语言中,实型常量有两种表示形式:十进制小数形式和指数形式。

(1)十进制小数形式。由正负号、数码 0～9 和一个小数点组成,小数点前面和后面可以没有数字。例如,下面都是正确的实型常数。

```
.123        // 表示实数 0.123
-.123       //表示实数 -0.123
123.        //表示实数 123.0
0.          //表示实数 0.0,也可以写成.0,但不可以把两个 0 都省略
```

(2)指数形式。由十进制小数(或整数)与字母 e(或 E)组成。一般形式如下:

aEn

或

aen

其中,a 可以是十进制小数或整数,n 必须为十进制整数,整体表示数 $a\times10^n$。对于以指数形式表示的实型常量,要求字母 e(或 E)前面必须有数字,后面必须为整数。例如,2.1E5、3.7E7、－2.8E－2 等均是合法的实数,而 E7、2.7E 不是合法的实数,因为 E7 中字符 E 之前没有数字,2.7E 中字符 E 之后没有整数。

实型常量在不加任何后缀时,系统默认为双精度型。实型常量的后缀用大写字母 F 或小写字母 f 表示单精度型,用大写字母 L 或小写字母 l 表示长双精度 long double 型。

2. 实型变量

实型变量可以分为以下 3 类:

(1)单精度型。类型说明符为 float,在内存中占 4 字节,其取值范围的绝对值为 10^{-38}～10^{38},提供 6～7 位有效数字。

(2)双精度型。类型说明符为 double,在内存中占 8 字节,其取值范围的绝对值为 10^{-308}～10^{308},提供 15～16 位有效数字。

(3)长双精度型。类型说明符为 long double,在内存中占 10 字节,其取值范围的绝对值为 $10^{-4\,932}$～$10^{4\,932}$,提供 18～19 位有效数字。

实型数据的存储形式决定了它能提供的有效数字是有限的,有效数字位数以外的数字会被舍去,所以实型数据一般存在误差。

【例 2-2】 实型变量的定义与使用。

```
//FileName: chap2_2.c
#include <stdio.h>
int main( )
{
    float a,b;                    //定义 a、b 为单精度浮点型变量
    double d;                     //定义 d 为双精度浮点型变量
    a = 3.56; b = 12345.678;
    d = 12345.6789;
    printf("%f,%f,%f\n",a,b,d);
    return 0;
}
```

程序运行结果如下：

```
3.560000,12345.677734,12345.678900
```

需要说明的是，变量 b 为 float 类型，有效数字为 6～7 位，故为 b 赋值为 12 345.678，输出时却显示 12 345.677 734，前面 7 位数字是有效的，后面出现误差。

2.3.4 字符型常量与变量

计算机中处理的数据不仅是整数、实数这样的数值，还包括字符型数据。在 C 语言中字符型数据包括字符和字符串两种。字符型数据在内存中是以字符的 ASCII 码值的二进制形式存储的，一个字符占用一个字节。

例如，字符 'a' 在内存中的存储形式如图 2-4 所示。

0	1	1	0	0	0	0	1

图 2-4 字符 'a' 的内存存储形式

1. 字符常量

C 语言中的字符常量分为普通字符常量和转义字符两种：

(1)普通字符常量。普通字符常量是用单引号括起来的一个字符。该字符可以是数字、字母等 ASCII 字符集中除“'”和“\”之外的所有可显示字符。例如，'a'、'E'、'3'、'+'、'$' 等都是合法的字符常量。

(2)转义字符。转义字符是一种特殊的字符常量。转义字符以反斜杠“\”开头，后跟一个或几个字符。转义字符具有特定的含义，不同于字符原有的意义，故称“转义”字符。例如，'\n' 就是一个转义字符，表示换行。常用转义字符的具体含义见表 2-5。

表 2-5 常用转义字符及其含义

转义字符	含　义	ASCII 码值
\a	响铃(BEL)	7
\b	退格(BS)	8

续表

转义字符	含　义	ASCII 码值
\f	换页(FF)	12
\n	换行(LF)	10
\r	回车(CR)	13
\t	水平制表(HT)	9
\v	垂直制表(VT)	11
\\	反斜杠	92
\?	问号字符	63
\'	单撇号字符	39
\"	双撇号字符	34
\0	空字符(NULL)	0
\ddd	任意字符	1～3 位八进制
\xhh	任意字符	1～2 位十六进制

可将表 2-5 中转义字符的使用方法归纳为以下 3 种：

①反斜杠后面跟某些特定字符表示不可打印的控制字符和特定功能的字符。

②表示具有特定含义的单撇号、双撇号和反斜杠字符。

③反斜杠后面跟一个八进制或十六进制数表示任意字符，其中八进制或十六进制数是该字符对应的 ASCII 码。例如，'\102' 表示 ASCII 码为八进制 102 的字符 'B'。

【例 2-3】 转义字符的使用。

```
//FileName: chap2_3.c
#include <stdio.h>
int main( )
{
    printf("\101 \x42 C\n");
    printf("I say:\"How are you? \"\n");
    printf("\\C program\\\n");
    return 0;
}
```

程序运行结果如下：

```
A  B  C
I say: "How are you? "
\C program\
```

2. 字符变量

字符变量的类型说明符为 char。字符变量的定义与其他类型变量相同，如下所示：

```
char a,b;
```

每个字符变量被分配一个字节的内存空间。由于字符变量在内存中存放的是字符的 ASCII 码

值，所以也可以把它们看成整型量。字符型数据与整型数据之间的转换比较方便。字符数据可以参与算术运算，也可以与整型量相互赋值，还可以按照整数形式输出。

【例 2-4】 分析以下程序的执行结果。

```
//FileName: chap2_4.c
#include <stdio.h>
int main( )
{
    short int n = 97;  //字符 'a' 的 ASCII 码为 97
    printf("%d, %c, %d, %c\n", n, n, n + 1, n + 1);
    return 0;
}
```

程序运行结果如下：

```
97, a, 98, b
```

在【例 2-4】中，字符 'a' 的 ASCII 码为 97，变量 n 以整型数据方式输出时为 97，以字符方式输出时为 'a'，如图 2-5 所示。n+1 的结果为 98，字符 'b' 的 ASCII 码为 98，n+1 以整型数据方式输出时为 98，以字符方式输出时为 'b'。

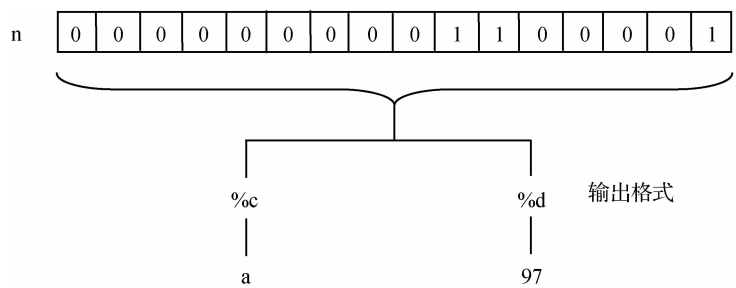

图 2-5 变量 n 的两种输出方式

3. 字符串常量

C 语言中的字符串常量是由一对双引号括起来的字符序列。例如，"C program"、"China"、"123.456"等都是合法的字符串常量。

字符串常量和字符常量是不相同的量，其区别如下：

(1)从表示形式上看，字符常量是由单引号括起来的，字符串常量是由双引号括起来的。

(2)从字符的个数上看，字符常量只能是单个字符，字符串常量可以包含 0 个或多个字符。

(3)有字符变量，但没有字符串变量。C 语言没有专门的字符串类型变量，而是使用字符型数组或字符型指针来存储字符串。

(4)字符常量在内存中占 1 字节，字符串常量在内存中的字节数为字符个数加 1。这是因为 C 语言规定，每一个字符串的末尾加一个字符串结束标志\0(ASCII 码为 0)。

例如，字符串"China"在内存中实际占 6 字节，如图 2-6 所示。

所以，字符常量 'a' 与字符串常量"a"虽然都只有一个字符，但两者截然不同，'a' 在内存中占 1 字节，而"a"在内存中占 2 字节。

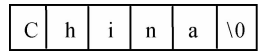

图 2-6　字符串常量"China"在内存中的存储

2.4　运算符和表达式

运算符是告诉编译程序执行特定算术或逻辑操作的符号,C 语言的内部运算符很丰富,除控制语句和输入/输出语句外,几乎所有基本操作都被作为运算符处理。C 语言有 3 种运算符:算术运算符、关系与逻辑运算符、位操作运算符。另外,C 语言还有一些特殊的运算符用于完成一些特定的任务。C 语言的运算符分类见表 2-6。

表 2-6　C 语言的运算符

运算符类型	运　算　符
算术运算符	+、-、*、/、%、++、--
关系运算符	>、<、==、>=、<=、!=
逻辑运算符	!、&&、||
位操作运算符	<<、>>、~、|、^、&
赋值运算符	=及其复合赋值运算符
条件运算符	?:
逗号运算符	,
指针运算符	*、&
求字长运算符	sizeof
特殊运算符	( )、[ ]、->、.

C 语言提供的运算符很多,学习时应该注意以下几点:

(1)运算符的功能。首先要记住运算符的功能。有的运算符可能有双重功能,当连接的运算对象不同时,功能也不同。例如,若有"int x;",则"&x"表示取变量 x 的地址,而"5&3"表示对 5 和 3 进行位运算。

(2)运算符的优先级。C 语言中的运算符优先级共分为 15 级,1 级最高,15 级最低。在表达式中,优先级较高的先于优先级较低的进行运算。运算符优先级相同时,则按运算符的结合性所规定的结合方向处理。

(3)运算符的结合性。C 语言中各运算符的结合性分为两种:左结合性(自左至右)和右结合性(自右至左)。例如,算术运算符是左结合性运算符,即先左后右;赋值运算符是右结合性运算符,即先右后左。

(4)运算结果及其表示。参与运算的数据类型不同,得到的结果就不同,因而表示方法和输出方法也不同。

按照一定的 C 语言语法规则,用运算符将常量、变量和函数连接起来组合而成的式子,就是 C 语言表达式。在 C 语言表达式中,还可以使用圆括号( )将需要优先计算的部分括起来。单个常量或变量可以看成 C 语言表达式的特例。任何表达式按一定规则计算后都会得到一个结果值,该结果具有相应的数据类型。

2.4.1 算术运算符及算术表达式

1. 算术运算符

算术运算符在很多计算机语言中都是最常用的，C语言的算术运算符可以分为基本算术运算符、自增和自减运算符及正负号运算符。

(1)基本算术运算符。基本算术运算符包括+(加)、-(减)、×(乘)、/(除)、%(取余)5种。

基本算术运算符都是双目运算符，即应有两个运算对象参与运算，如a+b、a-b、a×b、a/b、a%b等。基本算术运算符的优先级遵循“先乘除，后加减”的规则。×、/、%为同一级别，+和-为同一级别，且×、/、%的优先级别高于+和-。基本算术运算符的结合性为左结合。

使用算术运算符时应注意以下两个问题：

①如果有一个实型数参与运算，那么运算结果就是double类型，因为所有实数都按double类型进行运算。其中，/运算在参与运算的量均为整型时，结果也为整型。

②取余运算要求参与运算的量均为整型，运算的结果等于两数相除后的余数。

【例2-5】 /(除)和%(取余)运算符举例。

```
//FileName: chap2_5.c
#include <stdio.h>
int main( )
{
    printf("%d, %d\n",20/7, -20/7);
    printf("%f, %f\n",20.0/7, -20.0/7);
    printf("%d\n",100%3);
    return 0;
}
```

程序运行结果如下：

```
2, -2
2.857143, -2.857143
1
```

(2)自增和自减运算符。C语言中的自增运算符为“++”，自减运算符为“--”，都是单目运算符，具有右结合性。运算符“++”表示操作数加1，运算符“--”表示操作数减1。

自增和自减运算符可用在操作数之前，也可用在操作数之后。例如，“x=x-1”可写成“x--”或“--x”，但在表达式中这两种用法是有区别的。自增或自减运算符在操作数之前，表示在引用操作数之前就先执行加1或减1操作；自增或自减运算符在操作数之后，表示先引用操作数的值，之后再进行加1或减1操作。

【例2-6】 自增和自减运算符举例。

```
//FileName: chap2_6.c
#include <stdio.h>
int main( )
{
```

```
    int i,k;
    i=5; k=++i;                    //赋值时,i 先增 1,再将 i 的值赋给 k
    printf("k=%d,i=%d\n",k,i);
    i=5; k=--i;                    //赋值时,i 先减 1,再将 i 的值赋给 k
    printf("k=%d,i=%d\n",k,i);
    i=5; k=i++;                    //赋值时,先将 i 的值赋给 k,再将 i 增 1
    printf("k=%d,i=%d\n",k,i);
    i=5; k=i--;                    //赋值时,先将 i 的值赋给 k,再将 i 减 1
    printf("k=%d,i=%d\n",k,i);
    return 0;
}
```

程序运行结果如下:

```
k=6,i=6
k=4,i=4
k=5,i=6
k=5,i=4
```

在使用自增和自减运算符时,应注意以下两个问题:

①自增和自减运算符的优先级高于基本算术运算符。

②自增和自减运算符的操作数只能是变量,不能是常量和表达式。

(3)正负号运算符。正负号运算符是单目运算符,如-a、-b、-5、+8 等。

正负号运算符的优先级与自增和自减运算符同级,高于基本算术运算符。它的结合方向为自右向左。

举例如下:

```
int i,k;
i=5;
k=-i--;
```

表达式"-i--"相当于"-(i--)",因此 i 的最终值为 4,k 的最终值为-5。

2. 算术表达式

用算术运算符将运算对象连接起来,符合 C 语法规则,并能说明运算过程的式子,称为算术表达式。算术表达式的构成规则如下:

(1)数值型常量、数值型变量、数值型函数调用。

(2)+(算术表达式)、-(算术表达式)。

(3)++整型变量、--整型变量、整型变量++、整型变量--。

(4)(算术表达式)双目算术运算符(算术表达式)。

(5)有限次使用上述规则获得的运算式也是算术表达式。

算术表达式的类型可能是整型、单精度实型或双精度实型。由于 C 语言中其他类型表达式的值也是整型或实型,所以也可以当做算术表达式来使用。

2.4.2 关系运算符及关系表达式

关系运算实际上就是比较运算，即将给定的两个运算对象进行比较，判断比较的结果是否符合给定的条件。例如，a>b 中的>表示一个大于关系运算，如果 a=5，b=3，那么大于关系运算的结果为真，即条件成立；如果 a=2，b=3，那么大于关系运算的结果为假，即条件不成立。

1. 关系运算符

C 语言提供了 6 种关系运算符，见表 2-7。

表 2-7 关系运算符

关系运算符	含　义
<	小于
<=	小于等于
>	大于
>=	大于等于
==	等于
!=	不等于

在关系运算符中，<、<=、>、>=的优先级相同，==和!=的优先级相同，且前 4 个运算符的优先级高于后 2 个。关系运算符的优先级低于算术运算符。

2. 关系表达式

关系表达式是由关系运算符连接表达式构成的，具体构成如下：

```
表达式　关系运算符　表达式
```

其中，表达式主要是算术表达式，也可以是字符型数据或关系表达式、逻辑表达式、条件表达式、赋值表达式或逗号表达式等。

关系表达式的值为逻辑值，有 true(用 1 表示)和 false(用 0 表示)两种。

例如，假设 a=3，b=4，c=5，则各种表达式的值如下所示：

(1)“a>b”的值为 false(0)。

(2)“(a>b)!=c”的值为 true(1)。因为“a>b”的值为 false(0)，而 0 不等于 c，所以该关系表达式成立，即为 true(1)。

(3)“a<b<c”的值为 true(1)。因为“a<b”的值为 true(1)，而 1 小于 c 成立，所以该关系表达式成立，即为 true(1)。

(4)“(a<b)+c”的值为 6。因为“a<b”的值为 true(1)，所以 1+5=6。

2.4.3 逻辑运算符及逻辑表达式

C 语言提供逻辑运算符，逻辑运算的结果为 true(真)或 false(假)。

1. 逻辑运算符

表 2-8 列出了 C 语言中的逻辑运算符。ANSI C 标准规定，参与逻辑运算的操作数可以不是逻辑值，该操作数非 0 时表示真，为 0 时表示假，但逻辑运算的结果只可以取逻辑值(真或假)，返回值为 1 或 0。

表 2-8　逻辑运算符

逻辑运算符	名称及含义	举　例
!	逻辑非(单目)	!x
&.&.	逻辑与(双目)	x&.&.y
||	逻辑或(双目)	x||y

逻辑与运算符(&.&.)和逻辑或运算符(||)都是双目运算符,逻辑非运算符(!)是单目运算符。其运算规则与数学中的定义相同,表 2-9 列出了逻辑运算的真值表。

表 2-9　逻辑运算的真值表

a	b	!a	!b	a&.&.b	a||b
非 0	非 0	0	0	1	1
非 0	0	0	1	0	1
0	非 0	1	0	0	1
0	0	1	1	0	0

从表 2-9 可以看出:

(1)对于逻辑与运算,当 a 和 b 同时为真时,a&.&.b 的值为真,否则为假。

(2)对于逻辑或运算,当 a 和 b 同时为假时,a||b 的值为假,否则为真。

(3)对于逻辑非运算,就是对操作数进行取反操作。

逻辑运算符中 &.&. 和||的优先级低于关系运算符,!的优先级高于算术运算符。

2. 逻辑表达式

用逻辑运算符和圆括号将操作数连接起来的、符合 C 语言语法规则的式子称为逻辑表达式。具体构成如下:

```
单目逻辑运算符　表达式
```

或

```
表达式　双目逻辑运算符　表达式
```

其中,表达式主要是关系表达式,也可以是字符型或算术表达式、条件表达式、赋值表达式、逗号表达式等。

需要强调的是,在求解逻辑表达式时,并不是所有逻辑运算符都要被执行,当表达式的运算结果能够确定时,运算过程将立即终止,后面的部分将不予执行。这种现象被称为逻辑运算符的短路现象,也被称为懒惰求值法。具体情况如下:

(1)x &.&. y &.&. z。只有 x 为真(非 0)时,才需要判断 y 的值,只有 x 和 y 都为真的情况下才需要判断 z 的值。因此只要判定 x 为假,系统就终止运算,此时整个表达式的值已经确定为假。

(2)x||y||z。只有 x 为假时,才需要判断 y 的值,只有 x 和 y 都为假才需要判断 z 的值。因此只要判定 x 为真,系统就终止运算,此时整个表达式的值已经确定为真。

下面举例说明逻辑表达式的应用。

【例 2-7】　假设 x=10,y=20,分别计算以下各逻辑表达式的值。

①!x。

②x && y。

③!x+5||10 % y>=x-10<y。

上述逻辑表达式的计算过程及结果如下：

①由于x=10，根据C语言规定，非0为真，因此表达式!x的计算结果为假，表达式返回0。

②操作数x和y的值都是非0值，根据运算符&&的运算规则，表达式x&&y的计算结果为真，表达式返回1。

③该表达式等价于((!x)+5)||(((10%y)>=(x-10))<y)，逻辑运算符||前面的表达式结果为真，根据逻辑运算符的短路现象，不需要计算后面表达式的值便知整个表达式的结果为真，表达式返回1。

【例2-8】 写出满足要求的合法的C语言逻辑表达式。

①用x表示0～9的字符。

②x和y都是大于0的数。

③判断x的取值范围在40～100之间，即40≤x≤100。

④判断某一年是闰年。

满足上述条件的合法的C语言逻辑表达式如下：

①x>=48 && x<=57。

②x>0 && y>0。

③x>=40 && x<=100。

④year % 4==0 && year % 100 !=0||year % 400==0。

2.4.4 赋值运算符及赋值表达式

赋值运算符(=)用于赋值运算，是C语言中最基本的运算符，分为基本赋值运算和复合赋值运算。由=连接的式子称为赋值表达式。

1. 基本赋值运算

基本赋值运算符的符号为“=”，其作用是将右侧表达式的值赋给左侧变量，基本赋值运算符是双目运算符。例如，x=10表示执行一次赋值运算，把常量10赋给变量x。

由赋值运算符将一个变量和表达式连接起来的式子称为赋值表达式。这里要注意的是，赋值运算符的右侧可以是任意一个合法的C语言表达式，包括常量或者另一个赋值表达式，但左侧必须是一个变量，不可以是常量或表达式。例如，x=10和x=y=10都是合法的赋值表达式，但5=8和x+y=8都是不合法的赋值表达式。

ANSI C标准规定，赋值运算符的优先级低于算术运算符、关系运算符和逻辑运算符，其结合性为右结合。

【例2-9】 假设变量x为整型，计算以下各赋值表达式的值。

①x=y=10。

②x=10+(y=20)。

③x=10+(y=20)/(z=30)。

上述赋值表达式的计算过程及结果如下：

①根据赋值运算符的右结合性，先将10赋值给变量y，此时变量y的值为10，同时表达式“y=10”的值也为10，然后将表达式的值10赋给变量x，此时x的值为10，因此整个赋值表达式的值也为10。

②首先计算赋值表达式“y=20”的值，得到结果20，同时变量y被赋值为20，然后计算10+20的

值得到结果 30,再进行赋值运算,此时 x 的值为 30,整个表达式的值也为 30。

③先计算表达式“y=20”和“z=30”的值,得到结果分别为 20 和 30,然后计算表达式“10+20/30”的值得到结果为 10,再进行赋值运算,此时 x 的值为 10,整个表达式的值也为 10。

在进行赋值运算时,如果赋值运算符两侧操作数的数据类型不一致,系统会将右侧操作数的类型转换成左侧变量的数据类型,然后进行赋值运算。具体情况如下:

(1)左侧变量和右侧操作数都是整型或字符型时,根据左侧变量和右侧操作数所占字节数的不同进行相应处理。假设左侧变量和右侧操作数分别为 a 位和 b 位,则具体处理方法如下:

当 a=b 时,直接赋值即可。

当 a<b 时,仅将右侧操作数低端的 a 位赋值给左侧变量即可。

当 a>b 时,将右侧操作数赋值给左侧变量低端的 b 位,高 a-b 位的处理方式是:如果左侧变量是无符号数或正数,那么全部补 0,否则全部补 1。

(2)右侧操作数为实型,左侧变量为整型或字符型,这时将操作数变为整型(舍去小数部分),再赋值给左侧变量。例如,变量 x 为 int 型,则经过表达式“x=5.8”的赋值运算后,x 的值将是 5。

(3)右侧操作数为整型或字符型,左侧变量为实型,这时按照左侧变量的数据类型,将右侧操作数补足有效位后再赋值给左侧变量。例如,变量 f 为 float 型,则经过表达式“f='a'”的赋值运算后,f 的值将是 97.000 000。

2. 复合赋值运算

在赋值运算符“=”之前可以加上算术运算符或移位运算符构成复合赋值运算符。C 语言规定可以使用 10 种复合赋值运算符,分别为+=、-=、*=、/=、%=、<<=、>>=、&=、^=和|=。本节只介绍前 5 种,其他将在后续章节进行详细介绍。

对复合的赋值表达式求值时,先将该运算符右侧的操作数与左侧的变量进行指定的复合运算,然后将计算结果赋值给左侧的变量,并作为该赋值表达式的值。复合赋值运算符的优先级与赋值运算符相同,结合性也是右结合。

【例 2-10】 假设变量 x=10,y=20,计算各赋值表达式的值。

①x+=10。

②x*=y+20。

③x+=x--=x / 10。

上述复合的赋值表达式的计算过程及结果如下:

①根据复合的赋值表达式的求解规则,“x+=10”等价于“x=x+10”,先将变量 x 的值与操作数 10 相加,得到计算结果为 20,然后进行赋值运算,将 20 赋给变量 x,此时整个表达式的值为 20,变量 x 的值也是 20。

②该复合的赋值表达式等价于“x=x*(y+20)”,首先将表达式“y+20”的值与变量 x 的值 10 相乘,然后将结果 400 赋值给变量 x。此时整个表达式的值为 400,变量 x 的值也是 400。

③由于赋值运算符的右结合性,该表达式等价于“x=(x+(x=x-(x/10)))”。因此,先计算表达式“x=x-(x/10)”的值,得到结果为 9,此时变量 x 的值也是 9。然后计算表达式“x=x+9”的值,得到结果为 18,此时整个表达式的结果为 18,变量 x 的值也是 18。

根据实际问题构造表达式时,采用复合赋值运算符有两点好处:一是可以简化程序的书写,使程序精练;二是可以提高编译效率,产生质量较高的目标代码。

2.4.5 条件运算符及条件表达式

条件运算符(?:)是 C 语言中唯一的一个三目运算符,其目的是进行条件判断。

条件运算符的一般格式如下：

```
表达式1? 表达式2:表达式3
```

“表达式1”“表达式2”和“表达式3”的类型既可以是一个简单的表达式，又可以是由各种运算符组成的复合表达式。条件运算符的运算规则可以描述为：如果“表达式1”的值为非0，即逻辑真，那么运算结果等于“表达式2”的值；否则，运算结果等于“表达式3”的值，如图2-7所示。

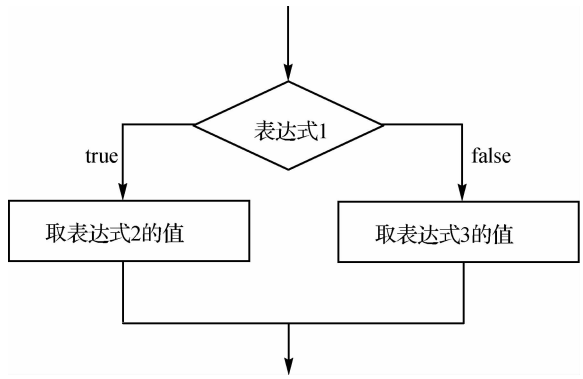

图2-7 条件运算符的运算规则

条件运算符的优先级高于赋值运算符，但低于关系运算符和算术运算符。其结合性为右结合。

【例2-11】 编写一个C语言程序，判断并输出用户输入的整数是奇数还是偶数。

```
//FileName: chap2_11.c
#include <stdio.h>
int main( )
{
    int n;
    printf("输入一个整数 n:");
    scanf("%d",&n);
    printf("%d是一个%s\n",n,(n%2==0?"偶数":"奇数"));
    return 0;
}
```

程序运行结果如下：

```
输入一个实数 n:5↙
5是一个奇数
```

从功能上讲，后面章节介绍的选择结构的if语句可完全实现条件运算符的功能，但在某些简单情况下，使用条件运算符可使程序更加简洁。

2.4.6 逗号运算符及逗号表达式

C语言提供一种特殊的运算符，即逗号运算符。其功能是用它将两个表达式连接起来组成一个表达式，称为逗号表达式。逗号运算符的结合性是自左至右，其优先级是最低的。

逗号表达式的一般形式如下：

```
表达式1,表达式2
```

逗号表达式的运算过程是:分别求两个表达式的值,并以表达式 2 的值作为整个逗号表达式的值。

【例 2-12】 逗号运算符举例。

```
//FileName: chap2_12.c
#include <stdio.h>
int main( )
{
    int a = 2, b = 4, c = 6, x, y;
    y = ((x = a + b), (b + c));
    printf("y = %d,x = %d\n", y, x);
    return 0;
}
```

程序运行结果如下:

```
y = 10,x = 6
```

在使用逗号运算符时应注意以下问题:

(1)逗号表达式一般形式中的“表达式 1”和“表达式 2”也可以是逗号表达式。例如,“表达式 1,(表达式 2,表达式 3)”形成了嵌套情形。因此,可以把逗号表达式扩展为以下形式:

```
表达式 1,表达式 2,…,表达式 n
```

整个逗号表达式的值等于“表达式 n”的值。

(2)并不是在所有出现逗号的地方都组成逗号表达式,在变量说明中,函数参数表中的逗号只是作为各变量之间的分隔符。

2.4.7 sizeof 运算符及 sizeof 表达式

sizeof 是 C 语言中的一种单目运算符。从形式上讲,它与后面章节要讲解的函数很相似,但并不是函数。将 sizeof 运算符与操作数组合在一起构成的式子称为 sizeof 表达式。sizeof 运算符用来获得一个数据或数据类型在内存中所占空间的字节数。sizeof 表达式的一般形式如下:

```
sizeof(表达式)
```

或

```
sizeof(数据类型名)
```

【例 2-13】 编写一个程序获得当前使用的编译系统所分配的数据类型、常量、变量和表达式所占内存字节数。

```
//FileName: chap2_13.c
#include <stdio.h>
int main( )
{
```

```
    int a = 2;
    printf("%d,", sizeof(int));
    printf("%d,", sizeof(long));
    printf("%d,", sizeof(float));
    printf("%d,", sizeof(double));
    printf("%d,", sizeof(char));
    printf("%d,", sizeof(6));
    printf("%d,", sizeof(a));
    printf("%d", sizeof(a + 6));
    return 0;
}
```

程序运行结果如下：

```
4,4,4,8,1,4,4,4
```

需要注意的是，同一种数据类型在不同的编译系统中所占空间不一定相同。例如，在基于 16 位的编译系统中，int 型数据占用 2 字节，但现在普遍使用的是基于 32 位的编译系统，int 型数据要占 4 字节。

到目前为止学过的 C 语言运算符的优先级与结合性如表 2-10 所示。

表 2-10 C 语言运算符的优先级与结合性

运 算 符	要求运算对象的个数	优 先 级	结 合 性
!,++,--,+,-,sizeof	1(单目运算符)	高 ↓ 低	右结合
×,/,%	2(双目运算符)		左结合
+,-	2(双目运算符)		左结合
<,<=,>,>=	2(双目运算符)		左结合
==,!=	2(双目运算符)		左结合
&&	2(双目运算符)		左结合
||	2(双目运算符)		左结合
?:	3(三目运算符)		右结合
=,+=,-=,*=,/=,%=	2(双目运算符)		右结合
,	2(双目运算符)		左结合

2.5 不同数据类型数据间的混合运算

很多情况下，在进行某种数值运算的过程中，会对操作数的数据类型进行转换，有些转换由系统自动进行，有些则由程序员人为指定。对于系统自动进行的类型转换通常要遵循一定的转换规则，如图 2-8 所示。

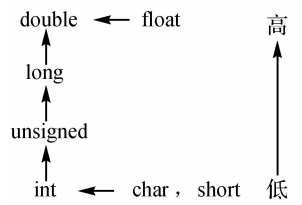

图 2-8　数值型数据在运算过程中的转换规则

图 2-8 体现了系统自动进行的两种类型转换：自动类型转换(也称隐式转换)和强制类型转换(也称显式转换)。

2.5.1　自动类型转换

自动类型转换发生在不同数据类型的量进行混合运算时，由编译系统自动完成。自动类型转换遵循以下规则：

(1)若参与运算的量的类型不同，则先转换成同一类型，然后进行运算。

(2)转换按数据长度增加的方向进行，以保证精度不降低。例如，int 型和 long 型运算时，先把 int 型转换成 long 型再进行运算。

(3)所有浮点运算都是以双精度进行的，即使仅含有单精度量运算的表达式，也要先转换成 double 型再进行运算。

(4)char 型和 short 型量参与运算时，必须先转换成 int 型。

【例 2-14】 计算各表达式的值，注意数据类型转换的过程。

①10－3.0/2。

②5L＋8。

上述表达式的计算过程及结果如下：

①操作数 3.0 为 double 型数据，因此操作数 10 和 2 会由系统转换为 10.0 和 2.0 参与运算，最终得到的结果为 double 型数据 8.500 000。

②操作数 5L 为 long 型数据，因此操作数 8 会由系统转换为 8L 参与运算，最终得到的结果为 long 型数据 13L。

2.5.2　强制类型转换

除赋值转换外，系统自动进行的数据类型转换都是由低精度类型向高精度类型的转换，如果需要将高精度类型的数值转换为低精度类型，必须由程序员人为指定，即强制转换。因此，强制类型转换须指定转换后的数据类型。

强制类型转换的一般形式如下：

```
(类型说明符)(表达式)
```

其功能是把表达式值的数据类型强制转换成类型说明符所指定的类型。例如，“(float)a”表示将变量 a 转换成 float 型，而“(int)(x＋y)”表示将表达式“x＋y”的值转换成 int 型。

使用强制类型转换时，需要注意以下问题：

(1)强制类型转换过程中有可能造成信息的丢失。例如，将实型数值转换为整型数值时会舍去小数部分。例如，表达式“(int)5.8f”的值是 5。

(2)如果被转换的数值在指定的结果类型中无法表示,那么虽然符合C语言的语法,这种转换也是没有意义的。例如,把52 769.8强制转换成int型,即(int)52 769.8,得到的结果为-12 767,一般来说没有意义。

(3)无论是强制类型转换还是自动类型转换,都只是为了本次运算需要而对变量的数据长度进行临时性转换,而不改变数据说明时对该变量定义的类型,即原变量不会发生任何变化。例如,假设float型变量f的值为5.6f,进行强制类型转换后,得到一个int型的数据5,但变量f的值仍然是5.6f。

(4)C语言将强制转换类型符看作单目运算符,单目运算符的优先级高于双目运算符。例如,表达式"(int)(5.6+8.8)×3"的值是42,而表达式"(int)5.6+8.8×3"的值是31.4。

【例2-15】 强制类型转换举例。

```
//FileName: chap2_15.c
#include <stdio.h>
int main( )
{
    float x;
    int i;
    x = 4.5;
    i = (int)x;   //将x临时强制转换成整型,离开本行x还是单精度型
    printf("x = %f,i = %d", x,i);
    return 0;
}
```

程序运行结果如下:

```
x = 4.500 000,i = 4
```

2.6 等级考试内容与真题解析

2.6.1 考试内容

全国计算机等级考试二级C语言程序设计考试大纲(2018年版)中针对本章的考试内容要求如下:

(1)C语言的数据类型(基本类型、构造类型、指针类型、无值类型)及其定义方法。

(2)C语言运算符的种类、运算优先级和结合性。

(3)不同类型数据间的转换与运算。

(4)C语言表达式类型(赋值表达式、算术表达式、关系表达式、逻辑表达式、条件表达式、逗号表达式)和求值规则。

2.6.2 真题解析

(1)以下选项中,能用作用户标识符的是(　　)。

A. void　　B. 8_8

C. _0_　　D. unsigned

解析 C 语言中规定,用户标识符必须由字母、数字和下划线组成,首字符不允许是数字,不能是关键字。故选 C。

(2)表达式“(int)((double)9/2)-9%2”的值是(　　)。

A. 0　　B. 3

C. 4　　D. 5

解析 考点主要涉及强制类型转换、/和%运算。本题答案为 B。

(3)若有定义语句“int x=10;”,则表达式“x-=x+x”的值是(　　)。

A. -20　　B. -10

C. 0　　D. 10

解析 表达式“x-=x+x”等价于“x=x-(x+x)”,所以表达式的值为-10。故选 B。

(4)设有定义“int x=2;”,以下表达式中值不为 6 的是(　　)。

A. x*=x+1

B. x++, 2*x

C. x*=(1+x)

D. 2*x, x+=2

解析 考点主要涉及复合的赋值运算和逗号表达式,逗号表达式的结果为最后一个表达式的值。本题答案为 D。

(5)以下不能正确表示代数式$\frac{2ab}{cd}$的是(　　)。

A. 2*a*b/c/d

B. a*b/c/d*2

C. a/c/d*b*2

D. 2*a*b/c*d

解析 D 选项表示的表达式是$\frac{2abd}{c}$。故选 D。

(6)C 语言源程序中不能表示的数制是(　　)。

A. 二进制　　B. 八进制

C. 十进制　　D. 十六进制

解析 C 语言源程序中能表示的进制是十进制、八进制和十六进制,不能表示二进制。故选 A。

(7)若有表达式“(w)?(--x):(++y)”,其中与 w 等价的表达式是(　　)。

A. w==1　　B. w==0

C. w!=1　　D. w!=0

解析 考点是当作为表达式时,w 与 w!=0 的值等价,!w 与 w==0 的值等价。故选 D。

(8)若有定义语句“int a=5;”,则表达式 a++的值是_____。

解析 自增运算符++在变量名后面,表示先取变量当前的值参与运算,然后变量自身的值再增 1。本题答案为 5。

(9)若有语句“double x=17; int y;”,当执行“y=(int)(x/5)%2;”之后 y 的值为_____。

解析 考点主要涉及强制类型转换、/和%运算。本题答案为 1。

2.7 习题

(1)求下列表达式的值。

①3.5+1/2。

②设“int x=18,k=14;”,表达式为 x%=k-k%5。

③(int)((double)(5/2)+2.5)。

④设 x=2.5,a=7,y=4.7,表达式为 x+a%3*(int)(x+y)%2/4。

⑤设 a=2,b=3,x=3.5,y=2.5,表达式为(float)(a+b)/2+(int)x%(int)y。

⑥设 a=2,b=5,表达式为 a++,b++,a+b。

⑦a 为 int 类型,且其值为 3,表达式为 a+=a--=a*a。

⑧x=y=6,x+y,x+1。

⑨x=(y=6,y*2,y+1)。

(2)编写程序:输入一个矩形的长和宽,计算该矩形的面积。

(3)编写程序:输入半径的值,计算并输出球的体积。

(4)编写程序:输入一个三位整数,求其百位、十位、个位上的数字,并求出各位数字之和及各位数字之积。