

## 第 1 章

# C 语言概述

通过本章的学习，读者应达成以下学习目标。

**知识目标** ▶ 了解 C 语言的发展历史和特点，理解 C 语言的基本概念，了解 C 语言程序的基本结构。

**能力目标** ▶ 理解程序从编辑、编译、连接到运行的工作原理，熟悉 Visual C++ 6.0 集成开发环境的各项功能，能够使用 Visual C++ 6.0 集成开发环境运行简单的 C 语言程序。

**素质目标** ▶ 培养学生对程序设计的兴趣，培养精益求精的科学素养。

## 1.1 初识 C 语言



语言是人与人之间交流的工具。程序设计语言是人与计算机交流的工具，C 语言是其中的一种。程序是使用程序设计语言编写出的一些语句序列，是人和计算机交流的方式。

### 1.1.1 C 语言的发展史

#### 1. 第 1 阶段：A 语言

C 语言的发展颇为有趣，真要寻根探源，它的原型还是 ALGOL 60 语言，也称 A 语言。

1960 年国际信息处理联合会（IFIP）设计了一种面向问题的结构化程序设计语言——ALGOL 60 语言。但它不能直接对硬件进行操作，不宜用来编写系统程序。

#### 2. 第 2 阶段：CPL 语言

1963 年，英国剑桥大学在 ALGOL 60 基础上推出 CPL（combined programming language）语言，该语言比较接近于硬件，但规模比较大，难以推广使用。

#### 3. 第 3 阶段：BCPL 语言

1967 年，英国剑桥大学的马丁·理查德（Martin Richards）对 CPL 语言进行了简化，推出了 BCPL（basic combined programming language）语言。

#### 4. 第 4 阶段：B 语言

1969 年，美国贝尔实验室的肯尼思·蓝·汤普森（Kenneth Lane Thompson）以 BCPL

语言为基础，又做了进一步简化，设计出了很简单的而且很接近硬件的 B 语言（取 BCPL 的第一个字母），并用 B 语言编写了第一个 UNIX 操作系统，在 DEC 公司的 PDP-7 小型机上运行。此时的 B 语言过于简单，功能有限。

### 5. 第 5 阶段：C 语言

1972 年 11 月，美国贝尔实验室的丹尼斯·麦卡利斯泰尔·里奇（Dennis MacAlistair Ritchie）在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母）。因此丹尼斯·麦卡利斯泰尔·里奇被称为 C 语言之父。C 语言既保持了 BCPL 和 B 语言的优点（接近硬件），又克服了它们的缺点（过于简单，数据无类型等）。

1973 年肯尼思·蓝·汤普森和丹尼斯·麦卡利斯泰尔·里奇合作，用 C 语言将 UNIX 操作系统进行了重写（即 UNIX 第 5 版），因此他们被称为 UNIX 之父。UNIX 操作系统由于使用了 C 语言而取得成功，几乎成为 16 位微机的标准操作系统，C 语言一诞生就由于编写 UNIX 操作系统的成功而引起了人们的关注。

1973 年之后，C 语言的发展相当迅速。1975 年 UNIX 第 6 版公布，1977 年又研制成功不依赖具体机器的 C 语言编译文本《可移植 C 语言编译程序》，推动了 C 语言在各种机型上的广泛应用。

1979 年，UNIX 第 7 版研制成功。丹尼斯·麦卡利斯泰尔·里奇和布莱恩·威尔森·柯尼汉（Brian Wilson Kernighan）合著了影响深远的名著 *The C Programming Language*，被称为标准 C。从而使 C 语言成为世界上应用最广泛的高级程序设计语言之一。

1983 年，美国国家标准协会（American National Standards Institute, ANSI）根据 C 语言问世以来各种版本对 C 语言的发展和扩充，制定了首个 C 语言标准草案，即 83 ANSI C。

1987 年，ANSI 公布了新标准，即 87 ANSI C。

1989 年，ANSI 又公布了新的标准，即 ANSI X3.159—1989，简称 C89。

1990 年，国际标准化组织（International Organization for Standardization, ISO）和国际电工委员会（International Electrotechnical Commission）接受 C89 作为国际标准 ISO/IEC 9899:1990，称之为标准 C，简称 C90。C89 和 C90 这两个标准只有细微的差别，因此，通常来讲 C89 和 C90 指的是同一个版本。

1995 年，ISO 对 C90 作了一些修订，简称 C95。

1999 年，C 语言标准又得到改进，这就是 ISO/IEC 9899:1999，简称 C99，该标准于 2000 年 3 月被 ANSI 采用。C99 标准相对 C89 有很多不同之处，例如变量声明可以不放在函数开头、支持变长数组等。

2011 年 12 月 8 日，ISO 与 IEC 正式发布 ISO/IEC 9899:2011，简称 C11，提高了 C 语言对 C++ 的兼容性，并增加了一些新的特性。

2018 年，ISO 与 IEC 正式发布 ISO/IEC 9899:2018，简称 C18。C18 没有引入新的语言特性，只对 C11 进行了补充和修正。本书的叙述以 C18 标准为主。

各软件厂商提供的 C 语言编译系统所实现的语言功能和语法规则略有差别，因此读者应了解所用的 C 语言编译系统的特点。

计算机编程语言总的来说可以分成机器语言、汇编语言和高级语言三大类。C 语言是目前国际上广泛流行的计算机高级语言集汇编语言和高级语言的优点于一身，既可用于编写系统软件，也可用于编写应用软件。因此熟练掌握 C 语言成为计算机开发人员的一项基本功。

## 1.1.2 C 语言的特点与应用领域

### 1. C 语言的特点

每一种语言都有自己的特点，C 语言也不例外，所以才有了语言的更替，有了不同语言的使用范围。C 语言主要有以下特点。

#### 1) 简洁紧凑、灵活方便

C 语言一共只有 37 个关键字，12 种控制语句，程序书写自由，主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。

#### 2) 运算符丰富

C 语言的运算符包含的范围广泛，共有 44 种运算符。C 语言把括号、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，可以在其他高级语言中难以实现的运算。

#### 3) 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、共用体类型等。能用来实现各种复杂的数据结构的运算。引入了指针概念，使程序效率更高。另外 C 语言具有强大的图形功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

#### 4) C 语言是结构化和模块化的语言

C 语言具有结构化的控制语句，控制程序流向，是一种结构化程序设计语言。这种结构化方式可使程序层次清晰，便于使用、维护以及调试。C 语言是以函数形式提供给用户的，用函数作为程序的模块单位，这些函数可方便地调用，便于实现程序的模块化。

#### 5) C 语法限制不太严格、程序设计自由度大

一般的高级语言语法检查比较严格，能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。C 语言程序生成代码质量高，程序执行效率高。

#### 6) C 语言可以进行底层开发

C 语言允许直接访问物理地址，可以直接对硬件进行操作，C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元，可以用来编写系统软件。

#### 7) C 语言适用范围大，可移植性好

C 语言有一个突出的优点就是适用于多种操作系统，如 DOS、UNIX、Windows、Linux 等系统，也适用于多种机型。

### 2. C 语言的应用领域

#### 1) 系统软件开发

C 语言可以用于编写系统软件，例如操作系统、编译器、解释器、调试器等，开发 Windows 系统内核、Linux 系统内核、苹果公司的 macOS。

#### 2) 应用软件开发

C 语言可以用于开发各种应用程序，包括微软办公软件 Office、金山办公软件 WPS；图形用户界面（GUI）应用程序；数据库系统如 Oracle、MySQL、SQL Server 等；功能强大的

数学软件 MATLAB 等都使用 C 语言开发。

由于 C 语言能够直接访问计算机硬件，因此可以编写出高效且可靠的应用程序。

### 3) 嵌入式底层开发

由于 C 语言具有高效且可移植的特性，使其成为嵌入式设备开发的首选语言。使用 C 语言可以编写出高效且可靠的嵌入式系统代码，例如，智能家居、自动驾驶、智能扫地机器人等。嵌入式实时操作系统 FreeRTOS、uCOS 和 VxWorks 等，都是用 C 语言开发的。

### 4) 游戏开发

C 语言具有图像处理能力，一些大型的游戏，环境渲染、图像处理、三维模型、二维图形、动画等使用 C 语言来处理。例如，跨平台游戏库 OpenGL、SDL 等都是用 C 语言编写的。

### 5) 网络程序开发

C 语言可以用于编写网络程序的底层代码和网络服务器端底层代码，例如 TCP/IP 协议栈的实现、HTTP 协议的实现、网络数据传输的实现等。

综上所述，C 语言是一种功能很强的语言。但是，它也有一些不足之处，即 C 语言语法限制不严格，虽然可使熟练的程序员编程更灵活，但安全性低；运算符丰富，完成功能强，但难记、难掌握。因此，学习、使用 C 语言不妨先学基础部分，先用起来，用熟练后再学不规范的语法规则，进而全面掌握 C 语言。

## 1.1.3 认识第一个 C 程序

下面通过一个例题来认识 C 程序。

【例 1-1】在计算机屏幕上输出一行字符。

```
#include <stdio.h>          /* 包含 stdio.h 的预处理语句 */
int main(void)              /* 主函数 */
{
    printf("This is a C program.\n"); /* 输出语句 */
    return 0;                /* 结束函数的执行 */
}
```

程序运行结果如下：

```
This is a C program.
```

说明：

(1) 程序第 1 行代码的作用是进行相关的预处理操作。其中字符“#”是预处理标志，用来对文本进行预处理操作，include 是预处理命令，它后面跟一对尖括号，表示头文件在尖括号内读入，stdio.h 就是标准输入/输出头文件，因为程序要使用输入输出函数，所以一般 C 程序的开头都要写上这样一行命令。放在源程序的最前面，用来提供输入输出函数的声明。

(2) 程序第 2 行代码声明了一个 main() 函数，该函数是程序的入口，main() 是函数名

字，表示“主函数”，英文圆括号中的内容是函数参数，这里 void 表示主函数没有参数，括号内的 void 也可省略不写。C99 之前 main() 函数的返回类型可定义为 void main()，main() 前面的 void 表示此函数是“空类型”，void 是“空”的意思，即执行此函数后不返回任何值。C99 规定 main() 函数的返回类型为 int 型。每一个 C 程序都必须有一个 main() 函数，C 语言的程序总是从 main() 函数开始执行，并且回到 main() 函数。

程序中以 /\* 开头到 \*/ 结尾中的所有内容表示 C 风格的注释。注释部分是既不参加编译，也不被执行的，仅仅为了增加程序的可读性和可维护性。C 风格注释的优势是方便跨行，即如果注释内容在一行内写不下，可以继续在下—行书写，只要内容在一对 /\* 和 \*/ 中间都被编译器当作注释来处理。注意斜线 (/) 和星号 (\*) 之间不能有空格，且注释不可以嵌套，即不能在一个注释中添加另一个注释。C99 允许使用 C++ 风格的注释，即单行注释符，以 // 开始，到本行末尾结束，且只能占—行。

(3) 程序第 3 行，函数体由花括号 “{” 表示开始。

(4) 程序第 4 行代码调用了—个用于格式化输出的 printf() 函数，在程序中 printf() 函数的作用是把括号内双引号之间的字符串按原样输出。“\n” 是换行符，即在输出 “This is a C program.” 后回车换行。这行程序末尾的分号，则是 C 语句结束的标志。

(5) 程序第 5 行 return 0; 的含义是结束函数的执行。

(6) 程序第 6 行，函数体由花括号 “}” 括起来表示程序结束。

## 1.2 C 语言的语法基础



### 1.2.1 C 语言字符集

C 语言的基本符号可分为 4 类，归纳如下。

(1) 英文字母：大小写各 26 个，共计 52 个。

(2) 阿拉伯数字：0、1、2、3、4、5、6、7、8、9 共 10 个数字。

(3) 下画线：\_。

(4) 特殊符号：通常由—两个符号组成，主要用来表示运算符。例如：

算术运算符：+、-、\*、/、%、++、--。

关系运算符：<、>、>=、<=、==、!=。

逻辑运算符：&&、||、!。

位运算符：&、|、~、^、>>、<<。

条件运算符：?:。

赋值运算符：=。

其他分隔符：( )、[ ]、{ }、.、,。

## 1.2.2 标识符

### 1. 标识符的概念

标识符就是用来标识变量名、符号常量名、函数名、数组名、数据类型名、宏以及文件名等有效字符序列。简单地说，标识符就是一个名字。

### 2. 定义规则

(1) 标识符只能由字母、数字和下画线 3 种字符组成，且第一个字符必须为字母或下画线。变量名、函数名等用小写字母表示，而符号常量全用大写字母表示，函数名和外部变量名由小于 6 个字符的字符串组成，系统变量由下画线“\_”起头构成。

(2) 在 C 语言中，大小写字母不等效。因此，a 和 A、I 和 i、Sum 和 sum 分别是两个不同的标识符。

(3) 用户自定义的标识符不能与 C 语言的保留字（关键字）同名，也不能和 C 语言库函数同名。

(4) 标识符应当直观且可以拼读，让读者看了就能了解其用途。标识符最好采用英文单词或其组合，不要太复杂，且用词要准确，便于记忆和阅读。切忌使用汉语拼音来命名。

(5) 在 C 语言中，标识符的长度可以是一个或多个字符，但是只有前面 32 个字符有效，即系统能识别的标识符最大长度为 32。

有的系统取 8 个字符，假如程序中出现的变量名长度大于 8 个字符，则只有前面 8 个字符有效，后面的不被识别。

例如，下面的 3 个变量名将被当作同一个标识符处理：

student\_name、student\_number、student\_sex。

因此，为了程序的可移植性及程序阅读方便，建议标识符的长度不要超过 8 个字符。

(6) 尽量避免名字中出现数字编号，如 value1、value2 等，除非逻辑上需要编号。

例如，下面为合法的标识符：

year、month、student\_name、sum2\_3、PRICE。

下面为不合法的标识符：

M.D.Jones、#a、\$432、3b、?c、a=b。

## 1.2.3 保留字

保留字又称为关键字，就是具有特定含义的标识符，用户不能用作自定义标识符。C 语言提供的标准保留字有 37 个，另外还有特定字 7 个。

### 1. 标准保留字

(1) 与数据类型有关的保留字有 14 个，包括 char、int、float、double、signed、unsigned、short、long、void、struct、union、typedef、enum、sizeof。

(2) 与存储类别有关的保留字有 4 个，包括 auto、extern、register、static。

(3) 与程序控制结构有关的标准保留字有 12 个，包括 do、while、for、if、else、switch、case、default、goto、continue、break、return。

(4) 其他的标准保留字有 2 个, 包括 `const`、`volatile`。

(5) C99 标准新增 5 个保留字, 包括 `inline`、`restrict`、`_Bool`、`_Complex`、`_Imaginary`。

## 2. 特定字

特定字有 7 个, 包括 `#include`、`#define`、`#ifdef`、`#ifndef`、`#undef`、`#endif`、`#elif`。

特定字用在预处理语句中, 虽然不是保留字, 但人们习惯将它们看成保留字, 并赋予了特定的含义。

## 1.3 设计简单的 C 程序



一个完整的计算机程序通常具备三个功能, 即输入数据、数据运算和输出结果。在 C 语言中, 数据运算主要是由赋值语句完成的, 而 C 语言本身不提供输入输出语句。在 C 语言中, 数据的输入、输出则需要调用 C 编译系统提供的 `scanf()` 函数与 `printf()` 函数来实现。在系统学习 C 语言程序设计之前, 本节先简要介绍赋值语句和 `scanf()`、`printf()` 函数的使用, 以便能进行简单的程序设计, 方便以后各章内容的叙述。

### 1.3.1 赋值语句

#### 1. 赋值运算符

赋值符号“=”就是赋值运算符, 它的作用是将一个数据赋给一个变量。

例如:

`a=5`, 其作用是执行一次赋值操作 (或称赋值运算), 将常量 5 赋给变量 `a`。

`x=a/b`, 将表达式 `a/b` 的值赋给 `x`。

#### 2. 赋值表达式

由赋值运算符连接起来的式子叫作赋值表达式。其一般形式为:

左值表达式 = 右值表达式

左值表达式必须是变量, 右边可以是表达式, 赋值表达式的操作是先计算赋值号“=”右边表达式的值, 再赋给左边的变量, 即将数值存入变量的存储单元中。

例如:

`a=b=c=10` 等价于 `a=(b=(c=10))`

即先将常量 10 赋给变量 `c`, 然后再将 10 赋给变量 `b`, 最后将 10 赋给变量 `a`。

#### 3. 赋值语句

赋值表达式后加上“;”就构成了赋值语句。

赋值语句的格式为:

左值表达式 = 右值表达式;

例如:

`x=10;`

`x=2*y+3;`

都是合法语句。

```
3=x-2*y;
```

是不合法语句。因为 3 是常数，不是变量。

C 语言允许在同一个语句中进行反复赋值。

例如：

```
a=b=c=1;
```

其操作是将 1 赋值给 c 变量，将 c 变量的值 1 赋值给 b 变量，再将 b 变量的值 1 赋值给 a 变量。其作用等同于 a=1; b=1; c=1;。这在有些语言中是不允许的。

### 1.3.2 printf() 函数

#### 1. printf() 函数的调用格式

printf() 函数的调用格式为：

```
printf (格式控制, 输出表列);
```

printf() 函数的主要功能是按“格式控制”所指定的格式，从标准输出设备上输出“输出表列”中列出的各输出项。在 printf() 函数结尾加上“;”就构成了格式输出语句。

#### 2. printf() 函数格式说明

(1)“格式控制”是由双引号括起来的字符串，包括格式说明和普通字符，格式说明由 % 和其后的格式字符组成，用来指定输出数据的输出格式。不同类型的数据需要不同的格式说明符。表 1-1 列出了 printf() 函数常用格式字符。

表 1-1 printf() 函数常用格式字符

格式字符	输出类型	说明
d	整型数据	以带符号的十进制形式输出整数（正数符号不输出）
o	整型数据	以八进制无符号形式输出整数（不输出前导符 0）
x	整型数据	以十六进制无符号形式输出整数（不输出前导符 0x 或 0X）
u	整型数据	以无符号的十进制形式输出整数
c	字符型数据	以字符形式输出，只输出一个字符
s	字符型数据	以字符串形式输出
f	实型数据	以小数形式输出单、双精度数，隐含输出 6 位小数
e	实型数据	以标准指数形式输出单、双精度数，数字部分小数位数为 6 位

(2)“输出表列”由若干个变量或表达式组成，之间用逗号“,”隔开。例如：

```
printf("a=%d,b=%d",a,b);
```

【例 1-2】printf() 函数的使用。

```
#include <stdio.h>
int main(void)
{
    int a,b,c,d;
    a=10;
    b=20;
    c=50;
    d=25;
    printf("a=%d,b=%d,c=%d,d=%d\n",a,b,c,d);
    printf("a+b=%d,a-b=%d\n",a+b,a-b);
    printf("a*b=%d,c/d=%d\n",a*b,c/d);
    return 0;
}
```

程序运行结果如下：

```
a=10,b=20,c=50,d=25
a+b=30,a-b=-10
a*b=200,c/d=2
```

说明：

- (1) 程序第 4 行，是声明部分，定义 a、b、c 和 d 四个整型变量。
- (2) 程序第 5～8 行，赋值语句，分别将常量 10、20、50 和 25，赋给变量 a、b、c 和 d。
- (3) 程序第 9～11 行，格式输出语句。
- (4) 程序第 12 行，return 0; 的含义是结束函数的执行。

### 1.3.3 scanf() 函数

#### 1. scanf() 函数的调用格式

scanf() 函数的调用格式为：

scanf (格式控制，地址表列)；

scanf() 函数的主要功能是按所指定的格式从标准输入设备读入数据，并将数据存入地址表列所指定的存储单元中。在 scanf() 函数结尾加上“;”就构成了格式输入语句。

#### 2. scanf() 函数格式说明

(1) “格式控制”是由双引号括起来的字符串，仅包括格式说明部分，格式说明由“%”和类型说明符组成，用于指定输入数据的类型。scanf() 函数用到的格式字符如表 1-2 所示。

表 1-2 scanf() 函数用到的格式字符

格式字符	输入类型	说明
d	整型数据	输入十进制整数
o	整型数据	输入八进制整数
x	整型数据	输入十六进制整数
u	整型数据	输入无符号十进制整数
c	字符型数据	输入单个字符
s	字符型数据	输入字符串
f	实型数据	输入实数，可用小数形式或指数形式输入
e	实型数据	与 f 作用相同

(2)“地址表列”是由若干个变量的地址组成的，就是在变量名前加“&”，当变量地址有多个时，各变量地址之间用逗号“,”隔开。“地址表列”中的地址个数必须与格式参数的个数相同，并且依次匹配。例如：

```
scanf("%d%d%d",&a,&b,&c);
```

当从键盘输入数据时，输入的数值数据之间用间隔符（空格符、制表符或回车符）隔开，间隔符数量不限。最后一定要按 Enter 键，scanf() 函数才能接收从键盘输入的数据。

### 1.3.4 库函数和头文件

前面介绍的 printf() 函数和 scanf() 函数是 C 编译系统提供的库函数。库函数不是 C 语言的组成部分，而是由 C 编译系统提供一些非常有用的功能函数（C 库函数见附录 C），例如各种输入输出函数、数学函数、字符串处理等，可供用户在自己程序中直接调用。这些库函数的说明、类型和宏定义都分门别类地保存在相应的“头文件”（也称包含文件或标题文件）中。在 C 语言的编译系统中，提供了若干个“头文件”。“头文件”以 .h 为扩展名，且为文本文件。当使用系统提供的库函数时，只需在程序开始加上预处理语句：

```
#include <头文件> 或者 #include "头文件"
```

这条语句指明用 #include 命令将“头文件”中的内容包含到用户源程序中，使之成为源程序的一部分。

printf() 函数和 scanf() 函数是在头文件 stdio.h 中定义的，stdio 是 standard input & output 的缩写，h 是 head 的缩写，因此，在调用它们之前，只要在程序开始加上预处理语句：

```
#include <stdio.h> 或者 #include "stdio.h"
```

然后就可以在程序中使用 scanf() 函数和 printf() 函数实现输入 / 输出功能。

C 语言有丰富的库函数，应该尽量利用。

【例 1-3】计算任意立方体的表面面积和体积。

设一个立方体的三条边分别是 a、b、c，表面积为 s，体积为 v，程序代码如下：

```
#include <stdio.h>
int main(void)
{
    int a,b,c,s,v;           /* 定义三个边为 a、b 和 c，表面积为 s，
                             体积为 v*/
    scanf("%d%d%d",&a,&b,&c); /* 输入三条边的长度。*/
    s=2*(a*b+b*c+a*c);     /* 计算表面积。*/
    v=a*b*c;               /* 计算体积。*/
    printf("s=%d\n",s);    /* 输出表面积。*/
    printf("v=%d\n",v);    /* 输出体积。*/
    return 0;
}
```

程序运行结果如下：

```
1 2 3<Enter>
s=22
v=6
```

说明：

(1) 程序第 4 行代码是声明部分，定义三个边为 a、b 和 c，表面积为 s，体积为 v，均为整型变量。

(2) 程序第 5 行代码，当运行到 `scanf("%d%d%d",&a,&b,&c);` 时，系统停下等待用户将整型数据从键盘输入给变量 a、b、c。其中，`&a`、`&b`、`&c` 中的“&”是地址运算符，`&a` 的含义是变量 a 在内存中的地址。`scanf()` 函数按照 a、b、c 在内存中的地址 `&a`、`&b`、`&c` 给 a、b、c 赋值。

`scanf()` 函数中的“%d%d%d”表示按十进制整数形式输入数据。从键盘输入时，两个数据之间用一个或多个空格间隔，也可用 Enter 键、Tab 键间隔。不能用逗号间隔两个数据。

(3) 程序第 6 行是赋值语句，将表面积的值赋值给 s 变量。

(4) 程序第 7 行是赋值语句，将体积的值赋值给 v 变量。

(5) 程序第 8 行是格式输出语句，`printf()` 函数括号中双引号括起来的“s=”按原样输出，“%d”表示“以十进制整数类型”输出 s 变量的值，`printf()` 函数括号内逗号右边 s 是要输出的变量。

(6) 程序第 9 行，格式输出语句，输出 v 变量的值。

(7) 程序第 10 行，`return 0;` 的含义是结束函数的执行。

程序运行时，从键盘输入 1、2、3，其间用空格分隔，并按 Enter（回车）键终止。

## 1.4 C 程序的开发过程



### 1.4.1 C 程序的实现步骤

C 语言是一种编译型的程序设计语言，开发一个 C 程序要经过编辑、编译、连接和运行四个步骤，4 个步骤都正确无误，才能得出正确结果，如图 1-1 所示。其中带箭头的实线表示操作流程，带箭头的虚线表示操作所需要的条件和产生的结果。

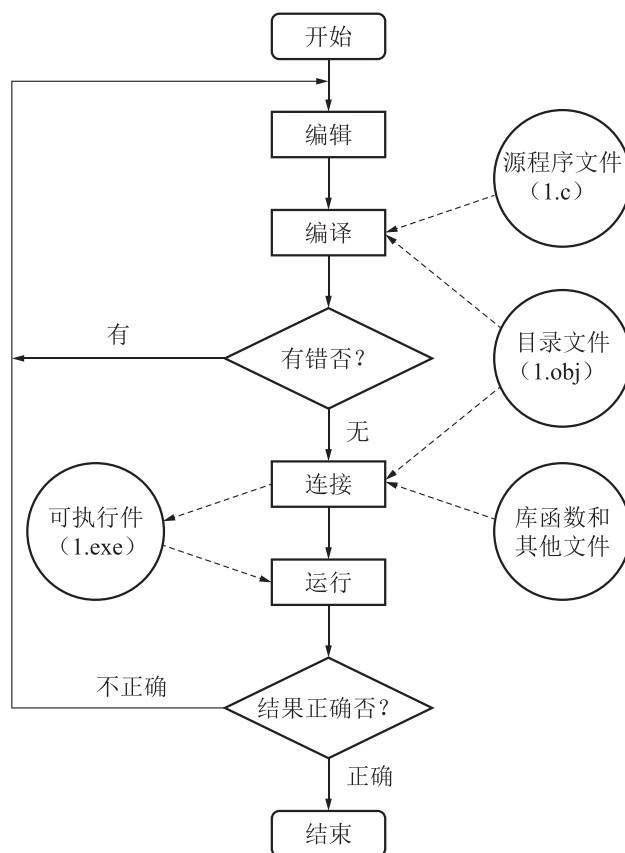


图 1-1 C 程序的开发过程

#### 1. 编辑

编辑是指 C 语言源程序的输入和修改，最后以文本文件的形式存放在磁盘上，文件名由用户自己选定，扩展名一般为 .c，这里以 1.c 为例。

#### 2. 编译

计算机硬件不能直接执行源程序，必须将源程序翻译成二进制目标程序，翻译工作由编译程序完成，翻译的过程称为编译，编译程序自动对源程序进行句法和语法检查，当发现错

误时，就将错误的类型和在程序中的位置显示出来，以帮助用户修改源程序中的错误。修改有语法错误的源程序称为调试。重新编译未发现句法和语法错误，就自动生成目标程序文件 1.obj。

### 3. 连接

连接也称链接或装配，是用连接程序将编译过的目标程序文件和程序中用到的库函数以及其他目标程序文件连接装配在一起，形成可执行的程序文件 1.exe。

### 4. 运行

将可执行的程序文件 1.exe 运行，由计算机执行后，结果不正确，重复前面的步骤；结果正确，可获取程序的正确结果。

## 1.4.2 C 语言常用集成开发环境

程序设计语言一般都有其编译运行环境。运行环境一般包括代码编辑器、编译器、调试器和图形用户界面工具，集成了代码编写功能、分析功能、编译功能、调试功能。这种集成了编译、运行、调试等功能的软件套组称为集成开发环境 (integrated development environment, IDE)。目前 C 语言常用的集成开发环境有 Turbo C、Borland C++ Builder、Microsoft Visual C++ 6.0、Microsoft Visual C++.NET 以及 Code::Blocks 等。

### 1. Turbo C

Turbo C 是美国 Borland 公司的产品，目前最常用的版本是 Turbo C 2.0。Turbo C 2.0 是该公司 1989 年推出的。Turbo C 2.0 基于 DOS 环境，在进入 Turbo C 2.0 集成开发环境后，不能用鼠标进行操作，主要通过键盘选择菜单，使用不够方便。

### 2. Borland C++ Builder

Borland C++ Builder 是由 Borland 公司继 Delphi 之后推出的一款高性能集成开发工具，具有可视化的开发环境。

### 3. Microsoft Visual C++ 6.0

Microsoft Visual C++ 6.0 (以下简称 VC++ 6.0) 是 Microsoft Visual Studio 家族的成员，是 Microsoft 公司 1998 年推出的基于 Windows 平台的可视化集成开发环境。VC++ 6.0 不仅可以开发 C++ 程序，也可以开发 C 程序。

### 4. Microsoft Visual C++.NET

Microsoft Visual Studio.NET 是 Microsoft Visual Studio 6.0 的后续版本，是一套完整的开发工具集。在 .NET 平台下包含 Visual C++ 开发组件。

### 5. Code::Blocks

Code::Blocks (简称 CB) 是一个开放源码的全功能的跨平台 C/C++ 集成开发环境。Code::Blocks 由纯粹的 C++ 语言开发完成，它被设计成具有很强的扩展性和完全可配置性。

由于 C++ 是从 C 语言发展而来的，C++ 对 C 程序是兼容的，也就是说，一个 C 程序可以在 C++ 集成开发环境中进行调试和运行。本书主要是以 Microsoft Visual C++ 6.0 为 C 语言集成开发环境，因为它的功能完善，操作简单，界面友好，适合初学者开发使用。

## 1.5 在 VC++ 6.0 中开发 C 程序



在 VC++ 6.0 集成开发环境中，一个 C 应用程序被称为一个项目或工程（project），它是由应用程序中所需的所有文件组成的一个有机整体，一般包括源文件、头文件、资源文件等。项目被置于项目工作区的管理之下。一个项目工作区可以包含多个项目，甚至是不同类型的项目。这些项目之间相互独立，但共用一个项目工作区的环境设置。

VC++ 6.0 有英文版和中文版，二者使用方法相同，只是中文版在界面上用中文替换了英文。本节介绍的是 VC++ 6.0 中文版，为了能使用 VC++ 6.0 集成开发环境，必须事先在所用的微型计算机上安装 VC++ 6.0 系统。本书中的程序都是在 VC++ 6.0 环境下调试和运行通过的。

### 1.5.1 VC++ 6.0 集成开发环境简介

#### 1. VC++ 6.0 的启动

在 Windows XP 以上操作系统上成功安装了 VC++ 6.0 以后，双击桌面上的“Microsoft Visual C++ 6.0”快捷方式图标，进入 VC++ 6.0 集成开发环境，屏幕上出现 VC++ 6.0 的主窗口。

#### 2. VC++ 6.0 主窗口

VC++ 6.0 主窗口由标题栏、菜单栏、工具栏、项目工作区窗口、编辑窗口、输出窗口和状态栏组成，如图 1-2 所示。

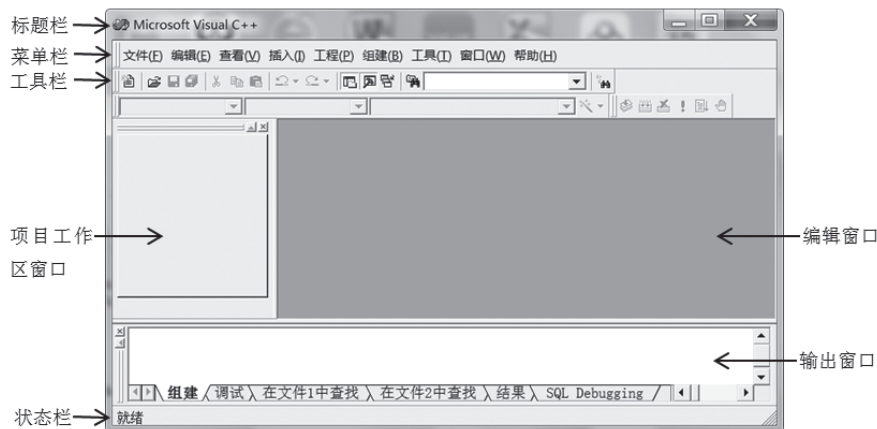


图 1-2 VC++ 6.0 主窗口

最上端的标题栏显示应用程序名和所打开的文件名（最大化时），标题栏的下面是菜单栏和工具栏，如果工具栏的左下方是工作区窗口，右下方是编辑窗口，再下面是输出窗口，主要用于显示项目建立过程中所产生的错误信息，最下方是状态栏，显示当前操作或所选命令的提示信息。

### 3. 开发环境的菜单栏

在 VC++ 6.0 集成开发环境界面中，菜单栏如图 1-3 所示，VC++ 6.0 集成开发环境大部分功能都是通过菜单来完成的。

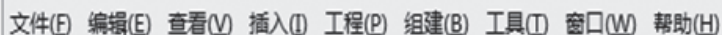


图 1-3 菜单栏

### 4. 开发环境的工具栏

工具栏是一种图形化的操作界面，具有直观和快捷的特点。工具栏是一系列工具按钮的组合。当鼠标停留在工具栏按钮上时，按钮凸起，主窗口底端的状态栏上显示出该按钮的一些提示信息。工具栏上的按钮通常和一些菜单命令相对应，提供一种执行常用命令的快捷方式。

### 5. 项目和项目工作区

一个 Windows 应用程序通常有许多源代码文件以及菜单、工具栏、对话框、图标等资源文件，这些文件都将纳入应用程序的项目中。通过对项目工作区的操作，可以显示、修改、添加、删除这些文件。项目工作区可以管理多个项目。

## 1.5.2 建立控制台应用程序

### 1. 创建项目工作区和项目

在 VC++ 6.0 中，程序在项目的管理之下，而项目则在工作区的管理之下。因此，开发一个 C 程序，首先要创建一个工作区和一个项目，其中，创建工作区和创建项目可以同时完成。

(1) 在 VC++ 6.0 开发环境中，选择“文件”|“新建”，弹出“新建”对话框，选择“工程”选项卡（默认），该选项卡中列出了 VC++ 6.0 可为用户创建的各种类型的应用程序，从中选择“Win32 Console Application”，创建一个基于控制台的项目。在“工程名称”下的文本框中输入新建项目名称，如 ex1\_1。在“位置”下的文本框中输入或选择该项目的存放路径，如 E:\MYCFILE\ex1\_1，并且选中“创建新的工作空间”单选按钮，如图 1-4 所示，最后单击“确定”按钮。



图 1-4 “新建”对话框下的“工程”选项卡

(2) 在弹出的“Win32 Console Application”对话框中，显示了四种项目类型，如图 1-5 所示，选择不同的选项，意味着系统会自动生成一些程序代码，为项目增加相应的功能。这里选中“一个空工程”单选按钮，表示生成一个没有任何源程序文件的空项目，再单击“完成”按钮。



图 1-5 “Win32 Console Application”对话框

(3) 在弹出的“新建工程信息”对话框中，显示将要创建的新项目的基本信息，如图 1-6 所示，单击“确定”按钮。



图 1-6 “新建工程信息”对话框

(4) VC++ 6.0 创建新项目，系统返回主窗口，并在文件夹 mycfile 下生成项目文件夹 ex1\_1 及该文件夹下的工作区文件等，并在项目工作区中显示与项目有关的信息（通过 Class View/File View 选项卡切换），如图 1-7 所示。

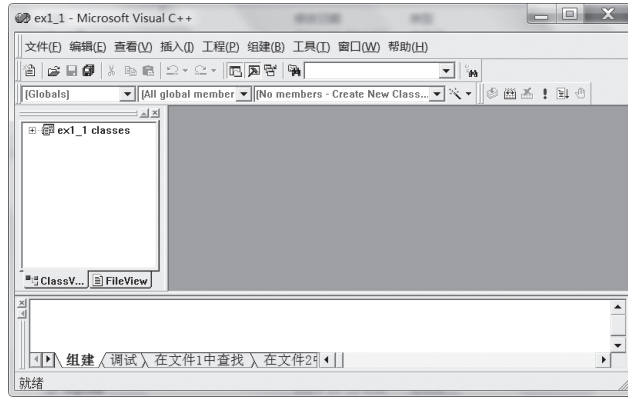


图 1-7 创建新项目后的主窗口

## 2. 创建和编辑程序源文件

创建的空白项目中没有任何文件，可添加各种类型的新文件到项目中。

(1) 在 VC++ 6.0 开发环境中，选择“文件”|“新建”，弹出“新建”对话框，选择“文件”选项卡。在“文件”选项卡中，列出了各种文件类型，从中选择“C++ Source File”，然后在确保右侧的“添加到工程”复选框被选中的情况下，在“文件名”下的文本框中输入新建源程序文件名，如 ex1\_1.c，如图 1-8 所示。单击“确定”按钮，系统返回主窗口，创建空的源程序文件 ex1\_1.c，将其加入项目中，并在文件编辑窗口中打开。

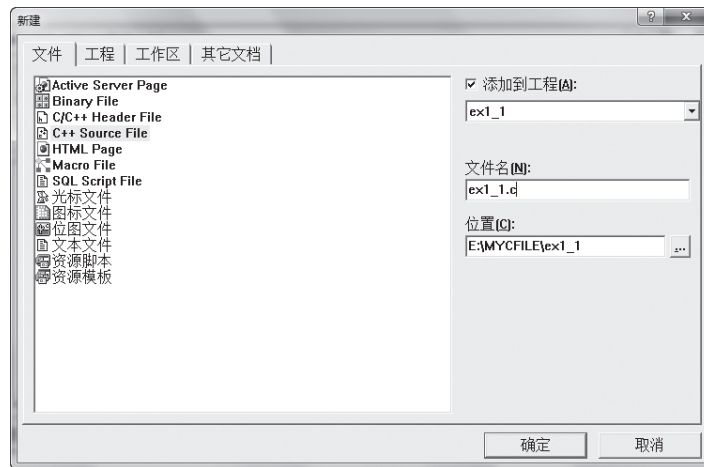


图 1-8 “新建”对话框下的“文件”选项卡

(2) 在编辑窗口中输入例 1-1 的源程序代码。输入结束，单击工具栏上的保存按钮，保存文件，如图 1-9 所示。

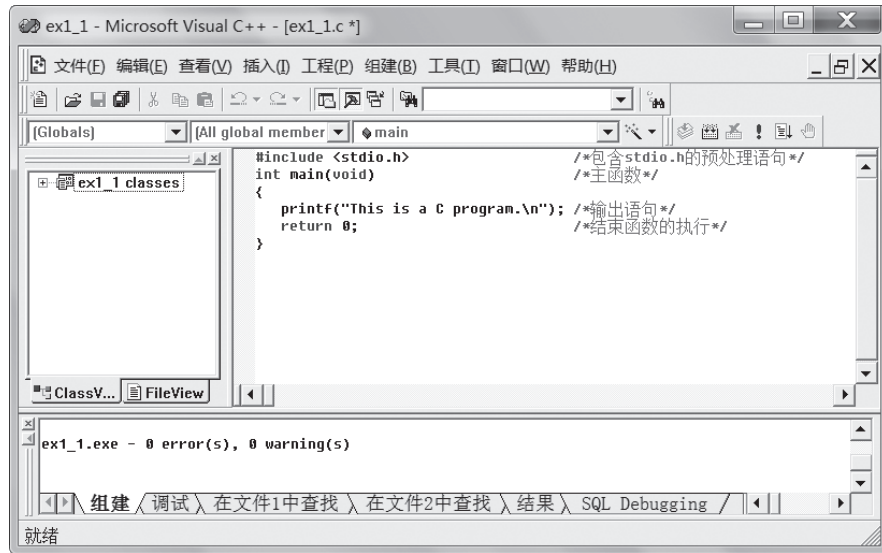


图 1-9 程序编辑窗口

### 3. 编译

在菜单栏中选择“组建”|“编译 [ex1\_1.c]”命令或按 Ctrl+F7 键，进行编译，如果显示错误信息，可以根据错误信息进行编辑、修改，并重新编译，直到没有错误信息为止。编译成功，输出窗口如图 1-10 所示，系统生成 ex1\_1.obj 目标文件，则可进行连接。

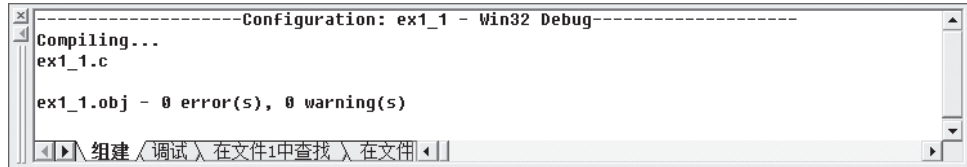


图 1-10 编译输出窗口效果

### 4. 连接

在菜单栏中选择“组建”|“组建 [ex1\_1. .exe ]”命令或按 F7 键，可将目标文件连接生成可执行文件 ex1\_1.exe，如图 1-11 所示。

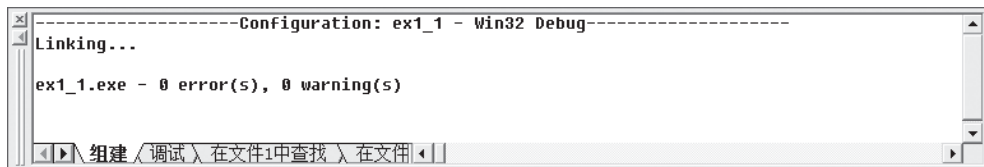


图 1-11 连接输出窗口效果

### 5. 运行

在菜单栏中选择“组建”|“! 执行 [ex1\_1.exe]”命令或按 Ctrl+F5 键，即可运行经编译、连接生成的可执行文件 ex1\_1.exe，运行结果显示在窗口屏幕上，如图 1-12 所示。

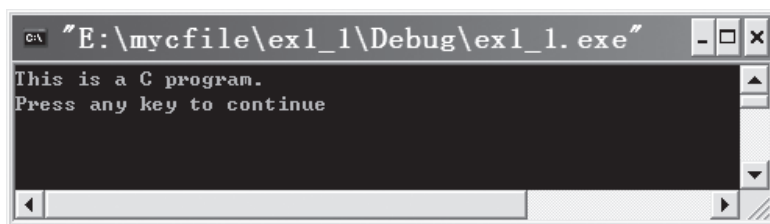


图 1-12 运行结果显示在窗口屏幕上

可以看到，在输出结果的窗口中第 1 行是程序的输出：

This is a C program.

第 2 行“Press any key to continue”并非程序指定的输出，而是由 VC++ 6.0 系统自动加上的一行信息，查看输出结果后，按任意键，输出窗口消失，返回到 VC++ 6.0 的主窗口。

## 1.6 程序案例

下面通过一个例题说明一下 C 语言程序的结构、特点及其设计风格。

【例 1-4】下列程序完成从算式  $1+2+3+\dots+n$  中找出一个其和大于 5000 的最小整数  $n$ 。

```
#include <stdio.h>
#define SUM_VALUE 5000
int sum(void)
{
    int total=0,n=0;
    while(total<=SUM_VALUE)
    {
        n++;
        total=total+n;
    }
    return(n);    /* 将所求出的 n 值返回给调用它的 main() 函数。*/
}
int main(void)    // 函数定义，函数名称为 main，称为主函数。
{
    printf("This least value of N is %d.\n",sum());
    return 0;
}
```

程序运行结果如下：

```
This least value of N is 100.
```

上述程序虽不复杂，但可以说明 C 语言程序的结构及其特点。现将其总结归纳如下。

### 1. 函数与主函数

C 语言中函数的概念类似于其他语言中的子程序概念，有所不同的是 C 语言中的主程序也被称为函数，并且被定义成名字为 `main()` 的主函数。因此，一个 C 语言程序是由一个或多个函数组成的，其中必须有一个主函数 `main()`。一个可执行的 C 语言程序总是从 `main()` 函数开始执行。

C 语言的函数有两类：一类是用户定义的函数，如 `sum()` 和 `main()` 是两个用户定义的函数，类型都是 `int` 型，其中 `main()` 是该程序的主函数。`main()` 后面的一对小括号是函数定义的标志，不能省略。用户定义的函数包括函数首部和函数体。另一类是系统提供的标准函数，在程序中所用到的系统标准函数是 `printf()`，它是系统提供的面向标准输出设备的格式输出函数。

### 2. 注释符

为了使程序易读，在 C 语言程序中，设置注释部分，帮助阅读和理解程序。注释符可以用来屏蔽程序中某行或某段代码的执行，用于程序调试。用作对代码的注释时，在相应代码的上一行或后面加“//”注册符；用作屏蔽某行代码的执行时，可在该行语句的前面加“//”注册符。而用作屏蔽某段代码的执行时，可将欲屏蔽的代码段放在“/\*”和“\*/”之间。注释越多，程序的可读性和可维护性也就越好。

### 3. 花括号“{”和“}”的作用

花括号在 C 语言程序中是用来构成函数体和复合语句（也叫程序块，即逻辑上相关的一组语句的集合）的分隔符。一个 C 语言程序中至少要有一对花括号，以表示程序体的开始和结束。

### 4. 程序语句

在 C 语言程序中，一条完整的语句必须以分号“;”结束。C 语言程序语句可分为以下 4 类。

#### 1) 说明语句

说明语句是用来说明变量的类型和初值的语句。例如，程序中的语句：

```
int total=0,n=0;
```

它说明 `total` 和 `n` 都是整型变量，并且初值都为 0。并列的标识符或项之间用逗号“,”分隔；两个关键词相邻时，中间用空格（至少一个）相间。

#### 2) 表达式语句

表达式语句是用来描述逻辑运算、算术运算或产生某种特定动作的语句。例如，程序中的语句：

```
n++;
total=total+n;
printf("This least value of N is %d.\n",sum());
```

这些都是表达式语句。`printf()` 函数是格式输出函数，此处是调用该函数完成输出功能。双引号内的字符串原样输出。`\n` 表示回车换行。