高等学校计算机应用规划教材

C 语言程序设计

(第三版)

刘国成　刘柏生　倪丹　编著

清华大学出版社

北　京

内 容 简 介

本书由编者根据20余年的教学经验编写而成，历经多次修正及改版，是适合C语言初学者的一部经典之作。本书内容循序渐进，浅显易懂，既有广度又不失深度，内容与案例以及讲解方式专为编程初学者而设计。该书从分析C语言程序的基本结构入手，介绍常量、变量、表达式和常用的输入/输出函数、流程控制、数组和字符串处理、函数、指针、编译预处理命令、结构体和共用体、文件、C语言高级程序设计和实验指导等知识点。书中以C89/C90标准为主线，兼顾C99和C11标准，示例程序都可在Visual C++ 6.0环境下编译和运行，每一章后面均附有习题，所涉及的内容全面，例题丰富。

本书既可作为高等院校相关专业的C语言课程教学用书，也可作为程序设计入门的参考书或培训教材。

本书配套的电子课件、实例源文件、习题答案可以通过http://www.tupwk.com.cn/downpage网站下载，也可以通过扫描前言中的二维码下载。

图书在版编目(CIP)数据

C语言程序设计 / 刘国成，刘柏生，倪丹 编著. —3版. —北京：清华大学出版社，2021.1
高等学校计算机应用规划教材
ISBN 978-7-302-57250-3

Ⅰ. ①C… Ⅱ. ①刘… ②刘… ③倪… Ⅲ. ①C语言－程序设计－高等学校—教材 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2020)第260553号

责任编辑：胡辰浩
封面设计：高娟妮
版式设计：孔祥峰
责任校对：成凤进
责任印制：丛怀宇

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：小森印刷霸州有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：22.25　　**字　　数**：569千字
版　　次：2014年6月第1版　　2021年3月第3版　　**印　　次**：2021年3月第1次印刷
定　　价：79.00元

产品编号：089160-01

前　言

欢迎学习 C 语言！C 语言作为一种语法清晰、功能强大、应用广泛的高级语言，长期以来被国内外高校定为程序设计的必修课。可以说掌握了 C 语言，就很容易掌握其他编程语言，如 Java、Python、C++、C#、PHP 等。但是很多初学者对学习 C 语言感到无从下手，究竟该怎样学习 C 语言？编者认为拥有一本合适的 C 语言教程是 C 语言入门的关键。

本书专为高等院校本科和专科相关专业的学生、软件开发人员和 IT 行业的专业人士而编写，具有如下特色：

(1) 本书由编者根据多年的教学经验编写而成，将 C 语言的知识体系做了精心编排。知识点涵盖了数据类型与表达式、流程控制、数组、函数、指针、编译预处理、结构体和文件等。授课教师可根据学生的情况对知识点进行调整或取舍。

(2) 每一章精心挑选具有代表性的例题，例题全部在 Visual C++ 6.0 环境下调试通过。

(3) 根据每章知识点精选了课后习题，读者应尽量独立完成课后习题，这对检验和巩固知识点大有益处。

(4) 本书以 C89/C90 标准为主线，兼顾 C99 和 C11 标准。为保证教材质量，编者查阅了大量的文献，研读了所有的 C 语言标准，尽量保证描述的准确性。

(5) 本书的实验指导章节是本书的最大特色，以成果导向教育(OBE)为理念，每个实验都设置了很多具体的实验任务，读者应尽可能完成所有实验任务，从而保证学习成果的达成度。

本书内容共分为 13 章，北华大学刘柏生老师编写了第 1~6 章，吉林工程技术师范学院刘国成老师编写了第 7~11 章，吉林工程技术师范学院倪丹老师编写了第 12~13 章。全书由刘国成老师统稿。

本书的出版得到了清华大学出版社相关同志的热切关心和大力支持。许多老师和读者也对本书的编写提出了诸多宝贵建议和修改意见，在此我们一并表示由衷的感谢。

由于编者水平有限，书中错误和不当之处在所难免，恳请读者批评指正。我们的邮箱是 huchenhao@263.net，电话是 010-62796045。

本书配套的电子课件、实例源文件、习题答案可以通过 http://www.tupwk.com.cn/downpage 网站下载，也可以通过扫描下方的二维码下载。

编　者

2020 年 10 月

目　录

第 1 章

C语言概述

C 语言是一种计算机程序设计语言。它由美国贝尔实验室的丹尼斯·里奇(Dennis Ritchie)于 1972 年推出。C 语言简单、可靠和易于使用，深受专业程序员和业余编程爱好者的喜爱。本章作为 C 语言的入门章节，首先介绍程序设计语言及其发展历史，C 语言的历史，C 语言的标准。然后通过剖析简单的 C 源程序，阐述 C 程序的基本构成，这也是本章的学习重点。同时本章还简单介绍 C 程序的运行步骤及常见的 C 语言集成开发环境。本章仅仅是 C 语言的初步介绍，大部分内容浅显易懂，具体和深入的技术细节将在后续章节中介绍。

1.1 程序设计语言及其发展

人们经常使用语言(如汉语、英语等)或文字来表达思想、交流和互通信息。人类相互交流信息所用的语言被称为自然语言，但是计算机目前还不能完全识别人类的自然语言。计算机能够识别的是计算机程序。计算机程序(Computer Program)是为完成某项特定任务而用计算机语言编写的一组指令序列。把解决某项任务的思路、方法和步骤最终落实为计算机程序的编写就是程序设计。用于书写计算机程序的语言称为程序设计语言(Programming Language)，它是人与计算机之间进行信息交流的工具。

程序设计语言的种类非常多，总的来说可以分为机器语言、汇编语言和高级语言这 3 大类。

1.1.1 机器语言

机器语言(Machine Language)是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言具有灵活、直接执行和速度快等特点。

1. 机器指令

机器指令是指挥计算机完成某一基本操作的命令，由硬件电路设计决定，因而也被称为硬指令。机器指令是由一组能被计算机接收的 0 和 1 组成的二进制代码。机器指令由操作码和地址码组成，规定了要求计算机完成的操作及其操作的对象(数据或存储单元地址)。

2. 指令系统

每台计算机所具有的特有的、全部指令的集合构成该 CPU 的指令系统。不同的 CPU 具有

不同的指令系统。

3. 机器语言程序

机器指令的集合构成了机器语言，用机器语言编写的程序就是机器语言程序。计算机所能识别的语言只有机器语言，但机器语言非常难以记忆和识别。人们在编程时，通常不采用机器语言，而采用汇编语言和高级语言。

1.1.2 汇编语言

汇编语言(Assembly Language)是面向机器的程序设计语言。在汇编语言中，用助记符代替机器指令的操作码，用地址符号或标号代替指令或操作数的地址，如此就增强了程序的可读性并且降低了编写难度。像这样符号化的程序设计语言就是汇编语言，因此亦称为符号语言。使用汇编语言编写的程序，机器不能直接识别，还要由汇编程序或者汇编语言编译器转换成机器指令。汇编程序将符号化的操作码组装成处理器可以识别的机器指令，这个组装的过程称为组合或者汇编。因此，有时候人们也把汇编语言称为组合语言。

1. 汇编指令

汇编指令是用助记符表示的机器指令，它与机器指令一一对应。

2. 汇编程序

计算机不能直接识别汇编指令，要让机器接收汇编指令还需要有一个将汇编指令翻译为机器指令的过程，这个过程称为汇编。汇编程序就是把汇编语言源程序翻译成机器语言程序的一种系统软件。IBM PC 中的汇编程序包括 ASM 和 MASM 这两种。ASM 称为小汇编程序，它只需要较小的存储区。MASM 称为宏汇编程序，它需要的存储区较大，但功能较强，且具有宏汇编能力。ASM 则不具备这种能力。

3. 伪指令

伪指令就是向汇编程序提供如何进行汇编工作的命令，也叫汇编控制命令。伪指令没有对应的机器指令，汇编时不产生机器码。

4. 汇编语言

汇编指令、伪指令、宏指令和汇编程序一起组成了汇编语言。汇编语言直接面向机器，用汇编语言编写的程序简洁、快速，常用于对运行速度要求较高的实时控制等场合。用汇编语言编写的用户程序称为汇编语言源程序。汇编语言的实质和机器语言是相同的，都是直接对硬件进行操作，但指令采用了英文缩写的标识符，更容易识别和记忆。而其所占用的存储空间和执行速度与机器语言相仿。

1.1.3 高级语言

上述的机器语言或汇编语言通常被称为低级语言(Low-level Language，LLL)，低级语言更接近于硬件，而高级语言(High-level Language，HLL)主要是相对于机器语言和汇编语言而言，它的第一个优点是以较接近人类语言的方式进行编程，用人们更易于理解的方式编写程序。第

二个优点是大多数高级语言的代码是可移植的，相同的代码可以在不同的硬件上运行。高级语言并不是特指某一种具体的语言，而是包括很多编程语言，如 BASIC 语言、C 语言、Python 语言等，这些语言的语法规范各不相同。

高级语言所编写的程序不能直接被计算机识别，必须翻译成机器代码，方式主要有解释和编译两种。

(1) 解释(Interpret)型。执行方式类似于人们日常生活中的“同声翻译”，一个称为解释器的程序读取每条程序语句，跟随程序流程，然后决定做什么并执行它。这种方式比较灵活，可以动态地调整、修改应用程序，如早期的 BASIC 语言便是采用这种方式。解释一个高级语言程序并运行的示意图如图 1.1 所示。

(2) 编译(Compile)型。编译是指在应用源程序之前，将程序源代码直接生成可以在机器上运行的机器码，机器码可以脱离其语言环境独立执行，使用比较方便且效率较高。但源程序一旦需要修改，就必须先修改源代码再重新编译，C 语言属于编译型。编译一个高级语言程序并运行的示意图如图 1.2 所示。

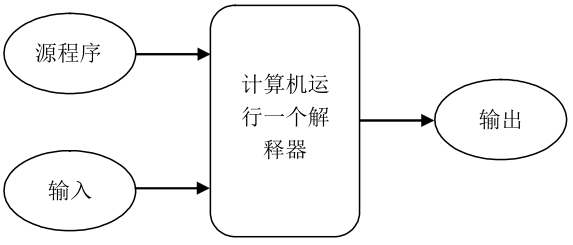

图 1.1　解释一个高级语言程序

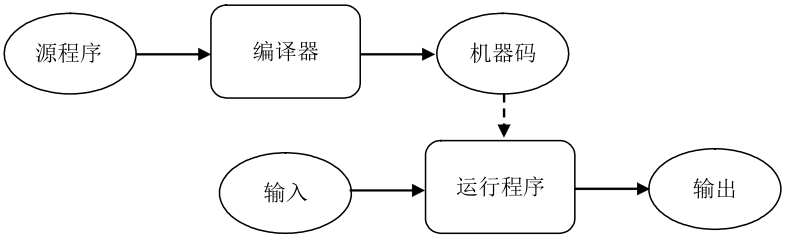

图 1.2　编译一个高级语言程序

1.2　C 语言的历史

C 语言是一种结构化程序设计语言。结构化程序设计方法主要由以下 3 种逻辑结构组成。

(1) 顺序结构。顺序结构是一种线性、有序的结构，它依次执行各语句模块。

(2) 选择结构。选择结构是根据条件成立与否选择程序执行的通路。

(3) 循环结构。循环结构是重复执行一个或几个模块，直到满足某一条件为止。

采用结构化程序设计方法，程序结构清晰，易于阅读、测试、排错和修改。由于每个模块执行单一功能，模块间联系较少，因此程序的编写比过去更简单，程序更可靠，而且增加了可

维护性，每个模块可以独立编写、测试。与大部分现代程序设计语言类似，C 语言来源于 ALGOL 语言。ALGOL 语言是最先使用块结构的程序语言。ALGOL 没有在美国得到普遍认可，但在欧洲却得到了广泛的应用。该语言给计算机科学界带来了结构化程序设计的概念。20 世纪 60 年代，计算机科学家科拉多·博姆(Corrado Bohm)、朱塞佩·雅科皮尼(Guiseppe Jacopini)和埃兹格·迪杰斯特拉(Edsger Dijkstra)使这一概念进一步大众化。

在 C 语言诞生之前还存在着一系列相关的程序语言。1967 年，马丁·理查兹(Martin Richards)开发了一种称为基本组合程序设计语言(Basic Combined Programming Language，BCPL)的计算机语言。肯·汤普森(Ken Thompson)在 1970 年开发了一种名为 B 的类似语言。B 语言是 UNIX 操作系统的第一个版本的开发语言。随后，在 1972 年，丹尼斯·里奇(Dennis Ritchie)设计了 C 语言，它继承了 ALGOL、BCPL 和 B 语言的许多思想，并加入了数据类型的概念以及其他功能强大的特性。由于 C 语言是与 UNIX 操作系统一起被开发出来的，因此它与 UNIX 有着很强的关联。

多年以来，C 语言主要用于科研环境，但最终随着多种商用 C 编译器的发布以及 UNIX 操作系统的不断流行，在计算机界开始获得广泛支持。今天，C 语言可以运行在多种操作系统和硬件平台下。

1980 年，比雅尼·斯特劳斯特鲁普(Bjarne Stroustrup)开始用一种新的语言工作，这种语言被称作“带类的 C 语言”。它增加了大量的新特性，全面改进了 C 语言，其中最重要的特性就是类。这种语言经过改进和扩充，最终成为 C++。

1.3 C 语言的标准

1. C89/C90/C95

1983 年，美国国家标准协会(American National Standard Institute，ANSI)开始制定 C 语言的标准，并于 1989 年 12 月通过，这个 C 标准被批准为 ANSI X3.159-1989，简称为 C89。

1990 年，国际标准化组织(International Organization for Standardization，ISO)采用了 ANSI C 标准(格式有变化)，即 ISO/IEC 9899:1990，该标准简称为 C90。C89 和 C90 基本可以认定为一个标准。

在经过 ANSI/ISO 标准化过程之后，C 语言规范在几年内保持相对稳定。1995 年，发布了新的标准 ISO/IEC 9899/AMD1:1995，简称为 C95，该标准纠正了一些细节并增加了对国际字符集的更广泛支持。

2. C99

20 世纪 90 年代末，对 C 标准进行了进一步修订，于 1999 年出版了 ISO/IEC 9899:1999，简称为 C99。C99 引入了一些新特性，包括内联函数、一些新的数据类型(包括 long long int 以及表示复数的复杂类型)、可变长度数组和灵活的数组成员，改进了对 IEEE 754 浮点的支持、对可变宏的支持以及对单行注释的支持(利用//符号)。C99 在很大程度上与 C90 向后兼容，但在某些方面更为严格，特别是缺少类型说明符的声明不再隐式假定为 int。

3. C11

2007 年，又开始了 C 标准的修订工作，该修订版非正式地称为 C1X，直到 2011 年 12 月 8 日正式发布为 ISO/IEC 9899:2011——C11 。C11 标准为 C 和库添加了许多新特性，包括泛型、宏、匿名结构、改进的 Unicode 支持、原子操作、多线程和边界检查函数。它还使现有 C99 库的某些部分成为可选，并提高了与 C++的兼容性。

4. C18

ISO/IEC 9899:2018 标准文档——C18 于 2018 年 6 月发布，是 C 编程语言的当前标准。它没有引入新的语言特性，只是对 C11 中的缺陷进行了技术性的修正和澄清。

1.4　C 语言的程序结构

1.4.1　简单的 C 语言程序剖析

学习一门新程序设计语言的唯一途径就是使用它编写程序。下面引入 C 语言的设计者布莱恩 •克尼汉(Brian Kernighan)和丹尼斯 •里奇(Dennis Ritchie)合著的 *The C Programming Language* 一书中的第一个示例程序，使用该程序打印出字符串“hello,world”。尽管这个编程练习很简单，但对于 C 语言的初学者来说，它仍然可能成为一大障碍。要实现这个目的，编程者必须先编写程序文本，然后成功地编译，并加载、运行，最后输出结果。掌握这些操作细节以后，其他的事情就比较容易了。

【例 1.1】 编写问候程序，输出字符串“hello,world”。

源程序：

```
#include <stdio.h>
main( )
{
    printf("hello, world\n");
}
```

运行结果：

```
hello,world
```

程序分析：

一个 C 语言程序，无论大小，都是由函数组成的。函数中包含一些语句，以指定所要执行的操作，本例中函数的名称为 main。通常情况下，C 语言并没有限制函数必须取一个什么样的名称，但 main 是个特殊的函数。main 函数称为主函数，每个程序都以 main 函数为起点开始执行，这就意味着每个程序都必须包含一个 main 函数，函数体须由{}括起来。

C 语言编译系统将一些常用的操作或计算功能定义成函数，如 printf、scanf、sqrt、fabs 等。这些函数称为标准库函数，其声明部分放在指定的以.h 为扩展名的头文件中。例如，存放标准输入/输出库函数声明的头文件名为 stdio.h，在使用系统库函数时须将对应的头文件包含进来。#include <stdio.h>是一个预处理命令，以#号开始，其功能是包含文件“stdio.h”。

在 printf("hello, world\n");语句中，printf 是一个用于打印输出的库函数，在本例中被主函数 main()所调用，用于打印双引号内包含的字符串。双引号内包含的字符序列叫作字符串或字符串常量，“hello,world\n”就是一个字符串，是 printf 函数的参数。“hello, world!”是原样输出的字符串序列，printf 函数不会自动换行，\n 是个换行符。程序在执行时若遇到它输出将换行。

例如，语句

```
printf("I see, \n I remember!");
```

其输出如下。

```
I see,
I remember!
```

printf("hello, world\n");语句最后以分号(;)表示该语句结束。

在继续讨论更多的示例之前，应注意很重要的一点：C 语言对字母是区分大小写的(即大小写敏感)。例如，printf 和 PRINTF 并不相同。

【例 1.2】 求两个整数 10 和 20 的和并输出结果。

源程序：

```
/*
功能：计算两个数的和，并输出
*/
#include <stdio.h>     /*  包含头文件 stdio.h   */
main( )
{
      int a, b, sum; /*  定义变量   */
      a=10; /*  给变量 a 赋整数值 10   */
      b=20; /*  给变量 b 赋整数值 20   */
      sum=a+b; /*  求和  */
      printf("sum=%d\n", sum); /*  输出 sum 的值  */
}
```

运行结果：

```
sum=30
```

程序分析：

在本程序中，/*...*/是注释，不是程序的必需部分，在程序执行时注释不起任何作用。注释的作用是增加程序的可读性，因此，适当地在程序中对相关语句进行注释，是一种良好的程序设计风格。C 语言的注释方法有以下两种。

(1) 块注释。

形式如下。

```
/*
注释内容
*/
```

跨多行的注释语句，适用于注释多行，“/*”和“*/”之间的内容为注释内容。

(2) 行注释。

形式如下。

```
/*  注释内容  */
```

/* ....注释内容..... */放在一行上，通常放在语句之后。

C99 标准支持“//”行注释(这个特性实际上在支持 C89 标准的很多编译器上已得到支持)。

形式如下。

```
//注释内容
```

作用范围是从“//”后面开始至本行结束。

示例代码如下。

```
int a, b, sum;    // 定义变量
```

注释可以出现在程序中的任何位置，注释中的任何内容都不会被计算机执行。

程序中的 int a, b, sum; 语句定义 a、b、sum 为 int 类型的变量，在 C 语言和其他大部分编程语言中，使用变量来存储数据。每个变量都由一个变量名来标识。每个变量必须有一种类型，用于表示它所存储的是哪种类型的数据。C 语言的基本数据类型分为整型、实型、字符型等。变量必须先定义，然后才能使用。

变量定义的一般形式：类型标识符 变量名列表；

变量定义后其初值一般是不确定的，不能直接使用，示例代码如下。

```
int a; printf("%d\n",a);          /* 错误!   */
int a; a=10;printf("%d\n",a);  /* 正确 */
```

在定义变量的同时，也可以对变量赋初值，称为变量的初始化。示例说明如下。

语句 int sum=0; 定义了 sum 为 int 类型变量，为 sum 赋初值 0。

printf("sum=%d\n"，sum)；语句输出 sum 的值，C 语言中的 printf 是个用得很普遍的命令，称为格式输出函数。其基本命令格式如下。

```
printf(格式控制, 输出列表);
```

其中，格式控制部分用双引号引起来，里面通常包含两种信息：一种是普通字符，普通字符按原样输出；另一种就是以“%”开头的格式说明，它的作用是将数据按指定的格式输出。例如，“%d”表示对应的输出值(即 sum 的值)以十进制整数形式显示，其余都是普通字符。所以程序执行后输出的结果如下：

```
sum=30
```

对应关系如图 1.3 所示。

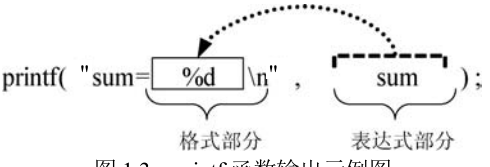

图 1.3　printf 函数输出示例图

如果在程序执行后想得到如下形式的输出结果：

```
10+20=30
```

则程序中的 printf 语句可改写如下。

```
printf("%d+%d=%d\n", 10,20,sum);
```

其中 10，20，sum 为输出列表，各表项之间用逗号分隔。由于有 3 个输出值，因此用 3 个格式说明符与之一一对应，如图 1.4 所示。

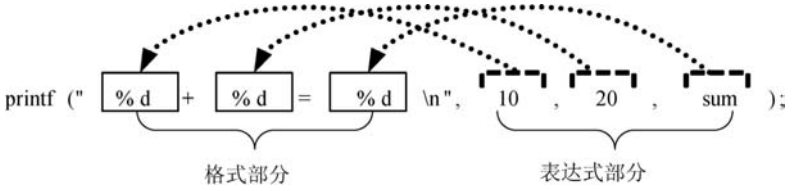

图 1.4　printf 函数输出示例图

例 1.2 中的程序只能计算 10 加 20 的和，因为程序中规定了 a 和 b 的值。如果要计算其他两个数的和，则需要修改程序。下面编写程序，使程序在运行时通过键盘操作输入需要求和的两个数，然后进行求和并输出结果。

【例 1.3】 求任意两个整数的和并输出结果。

源程序：

```
#include <stdio.h>
main( )
{
    int a, b, sum;          /*  定义变量  */
    scanf("%d", &a);    /*  输入第一个整数  */
    scanf("%d", &b);    /*  输入第二个整数  */
    sum=a+b;                /*  计算和  */
    printf("The sum of %d and %d is %d.\n", a,b,sum);  /*  输出和  */
}
```

运行结果：

```
33↙
55↙
The sum of 33 and 55 is 88.
```

程序分析：

在 C 语言中，要想获得从键盘输入的值，可以使用 scanf 函数。scanf 是与 printf 相对应的格式输入函数，其基本命令格式如下：

```
scanf(格式控制, 地址列表);
```

语句 scanf("%d", &a);表示以十进制整数的形式(由格式说明符“%d”指定)输入数据，存放到变量 a 对应的存储单元中，这样变量 a 的值就是刚刚从键盘输入的值。&是取地址运算符，&a 指变量 a 在内存中的地址。变量 a 和 b 及其所存储的数值如图 1.5 所示。

图 1.5　变量及其值

1.4.2　C 语言程序的基本结构

(1) C 语言程序是由函数组成的。一个完整的 C 程序可以由一个或多个函数组成，其中 main 主函数必不可少，且只有一个。C 程序执行时，总是从 main 函数开始，与 main 函数在整个程序中的位置无关，其他函数都是为 main 函数服务的。函数是 C 程序的基本单位，可以用函数来实现特定的功能，所以说 C 是函数式的语言。C 语言的函数包括系统提供的库函数(如 printf 函数)，以及用户根据实际问题编制设计的函数。

(2) 源程序中可以有预处理命令，预处理命令通常放在源文件或源程序的最前面。

(3) 每一条语句都必须以分号结尾，但预处理命令、函数头和右花括号“}”之后不加分号。

(4) 注释不是程序的必需部分，在程序执行时注释不起任何作用。注释的作用是增加程序的可读性，因此，适当地在程序中添加注释，是一种良好的程序设计风格。C 语言有块注释和行注释这两种注释方法。

(5) 在 C 语言中，虽然一行可写多条语句，一条语句也可占多行，但建议一行只写一条语句。

(6) 一般采用缩进格式书写程序，以提高程序的可读性和清晰性。

1.5　C 语言程序的运行

1.5.1　运行 C 语言程序的步骤

C 语言属于编译型的编程语言，如果要使 C 程序在计算机上执行，必须经过源程序的编辑、编译和连接等一系列步骤，最后得到可执行程序并运行，如图 1.6 所示。

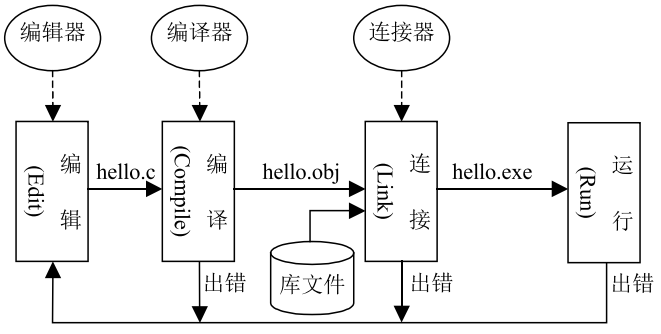

图 1.6　运行 C 程序的步骤

1. 编辑(Edit)

编辑是建立或修改 C 源程序的过程，并且该程序以文件的形式存储在磁盘上，C 源程序文件的扩展名为.c。

2. 编译(Compile)

C 语言编译器将 C 源程序转换为机器代码，生成目标程序。目标程序文件的扩展名为.obj。在 C 语言源程序的编译过程中，可以检查出程序中的语法错误。

3. 连接(Link，也称为链接)

编译生成的目标程序在计算机上还不能直接执行，还需将目标程序与库文件进行连接处理，连接工作由连接程序完成。经过连接后，生成可执行程序，可执行程序的扩展名为.exe。

4. 运行(Run)

C 源程序经过编译、连接后生成了可执行文件(.exe)。生成的可执行文件，既可在编译系统环境下运行，也可以脱离编译系统直接在操作系统下执行。

当编译时出现错误，说明 C 程序中有语法错误；若在运行时出现错误或结果不正确，说明程序设计上有错误(称为逻辑错误)，都需要修改源程序并重新编译、连接和运行，直至将程序调试正确为止。上述步骤中的第 2 步和第 3 步在一些如 Visual C++ 2010 的集成开发环境中进行了整合，源程序编辑完成后，一个“生成”操作就可以完成编译和连接工作。

1.5.2 集成开发环境

集成开发环境(Integrated Development Environment，IDE)可提供编程时所必需的工具。这些工具包括编辑器、编译器和调试器，它们集成在一个软件包内供编程人员使用。在早前的 DOS 环境下主要的 C 语言集成开发环境有 TC 2.0、Turbo C++ 3.0 和 Borland C++等。在 Windows 环境下主要有微软公司的 Visual C++ 6.0、Visual C++ 2010 及以上版本，截至目前，微软公司推出的 Visual Studio 2019 为 C 语言程序的最新版集成开发环境。

1.6 本书的约定

为了方便读者阅读本书以及完成书中源代码的上机实践，本节介绍一下书中示例代码、图示以及操作系统和开发环境的相关约定。

1.6.1 示例代码的约定

书中加入了大量的示例程序和部分源代码，为了方便读者阅读，源代码全部加入了深色填充底纹。本书用相同的字体表示输入的数据和计算机程序的输出结果，为区别于其他正文，这两部分内容也全部加入了深色填充底纹。

1.6.2 图示的约定

一般情况下，默认在每行输入的末尾都会按下 Enter 键，尽管如此，书中为了表示清楚，还是加了“↙”符号表示 Enter 键。另外，书中加入了一些流程图和软件界面的截图。

1.6.3　本书使用的 C 语言标准

考虑到兼容性，本书绝大部分内容采用 C89/C90 标准，较少部分内容会提及 C99 和 C11 标准。

1.7　本 章 小 结

C 语言是一种结构化的程序设计语言。C 程序主要由函数组成，main 函数是程序的入口。C 语言中的输入/输出由标准输入/输出函数来完成，其中 printf 函数完成输出功能，scanf 函数完成输入功能。C 语言属于编译型的编程语言，需要通过编辑、编译、连接后确认没有错误方可运行，目前 Windows 环境下常用的集成开发环境是 Visual C++ 6.0、Visual C++ 2010 及以上版本。

1.8　习　　题

1. C 语言程序由哪几部分组成？
2. 填空。

(1) 在 C 语言中，每条语句必须以_____结束。

(2) _____函数称为主函数。

(3) 存放标准输入/输出库函数声明的头文件名为_____。

3. 参照本章例题，编写一个 C 语言程序，输出以下信息。

```
*************************
Hello World!
*************************
```

4. 参照本章【例 1.3】，编写一个 C 语言程序，求任意两个整数的差并输出结果。
5. 指出下面程序中的错误。

```
#include (stdio.h)
main( )
{
    int x , y , s ;
    x=10
    y=20;
    s=x + y ;
    printf("s = %d\n , s );
}
```