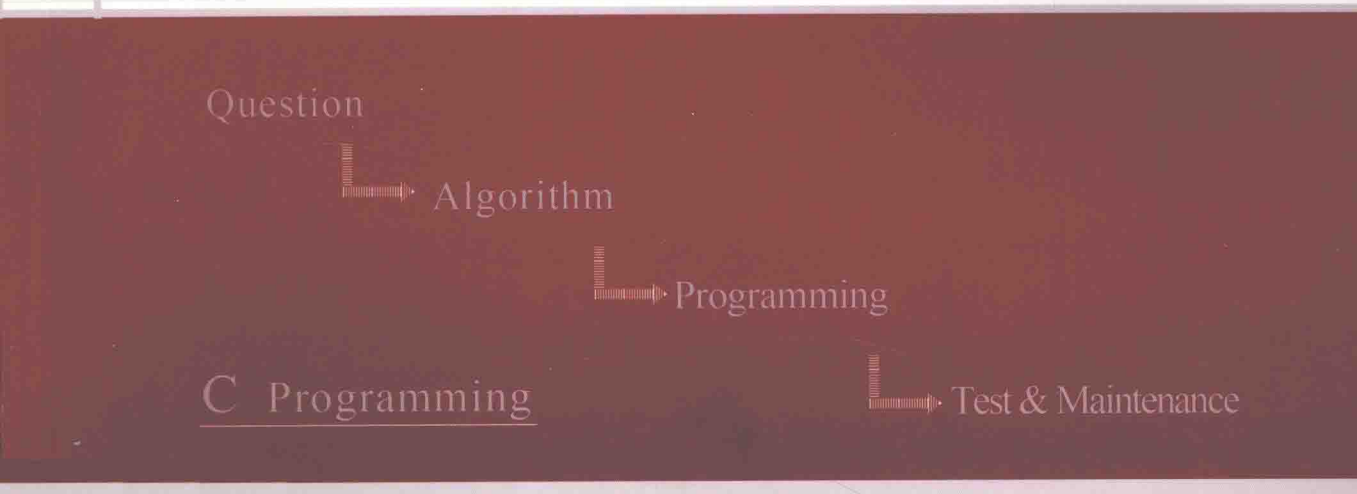
实用C语言程序设计

主　编　周百顺

副主编　刘　非　张文战



中国农业大学出版社
CHINA AGRICULTURAL UNIVERSITY PRESS

实用C语言程序设计

主　编　周百顺
副主编　刘　非　张文战

中国农业大学出版社
·北京·

内 容 简 介

本书以培养应用型人才为目标，旨在培养学生的结构化程序设计思想，锻炼自顶向下、逐步求精的分析问题和解决问题的能力，以及使用C语言完成相关程序设计的能力，同时注重程序设计规范的培养。本书将C语言的编程理念和语法相结合，共划分为10章进行讲解，内容编排合理，深入浅出，通俗易懂。每章内容相对独立完整，便于学生学习和理解，章节之间衔接流畅。每章均配有大量实际应用中的案例程序，并配有分析、讲解和相关习题。

图书在版编目(CIP)数据

实用C语言程序设计/周百顺主编. —北京：中国农业大学出版社，2014.7
ISBN 978-7-5655-0973-5

Ⅰ.①实… Ⅱ.①周… Ⅲ.①C语言-程序设计 Ⅳ.①TP312

中国版本图书馆CIP数据核字(2014)第104901号

书　　名　实用C语言程序设计
作　　者　周百顺　主编

责任编辑　王笃利　孙　勇　　**责任校对**　王晓凤　陈　莹
封面设计　郑　川
出版发行　中国农业大学出版社
社　　址　北京市海淀区圆明园西路2号　　**邮政编码**　100193
电　　话　发行部 010-62818525,8625　　读者服务部 010-62732336
　　编辑部 010-62732617,2618　　出　版　部 010-62733440
网　　址　http://www.cau.edu.cn/caup　　**E-mail** cbsszs@cau.edu.cn
经　　销　新华书店
印　　刷　北京时代华都印刷有限公司
版　　次　2014年7月第1版　2014年7月第1次印刷
规　　格　787×1 092　16开本　17.5印张　435千字
定　　价　40.00元

前　言

C语言是一种被广泛使用的结构化程序设计语言，也是软件开发人员从事软件开发工作的重要工具，具有与计算机底层结合紧密、执行效率高等特点，深受编程爱好者的喜爱。C语言程序设计课程一直是高等院校计算机及相关专业的核心专业基础课，对于培养学生的程序设计能力具有重要作用。

本书作者曾经在企业从事软件开发工作多年，积累了丰富的使用C语言进行软件开发的实践经验，进入高校后从事教学工作，讲授C语言程序设计课程。希望能够借助此书与读者分享C语言的学习经验和体会，帮助初学者开启程序设计的大门，为后续计算机相关知识和理论的学习打下良好的基础。

本书具有如下特色：

(1)内容编排合理、整体性强，讲解深入浅出。本书在内容安排上，既考虑到了相关知识的顺序性和依赖关系，又确保每个章节相对独立完整，使得读者在阶段性学习过程中仍能够从整体上把握程序设计的各个阶段。内容讲述过程中，尊重人们对知识的认知规律，通过类比等方法，深入浅出，将抽象的原理具体化、形象化，便于初学者掌握。

(2)注重结构化程序设计思想的讲解和能力培养。结构化程序设计以过程为中心，强调功能分解和模块化设计，采用自顶向下、逐步求精的方法进行问题分析和程序设计。本书将自顶向下、逐步求精的思想贯穿始终，重视程序的函数分解，培养学生使用结构化思想去分析问题和解决问题的能力。

(3)程序示例更贴近实际应用，实用性强，有助于提升学生的学习兴趣。本书为每章都配备了大量精选的程序实例辅助教学，在案例的选择上尽可能贴近实际应用，减少纯数学问题的案例，使读者能够更为直观地感受到知识的实用价值，同时重视从问题到程序，从理论到实际应用的过程讲解，提升学习兴趣。

(4)善于归纳总结。书中给出了很多一般性的程序设计原则和实战经验，供读者参考。包括对复杂程序进行函数分解的一般原则、根据问题的描述编写函数定义的原则、C语言中指针的主要应用领域等。

本书共分10章，第1章讲述了计算机的软硬件基础知识和相关工作原理，给出了程序设计语言的发展过程和高级语言的实现方法，介绍了C语言的历史和使用C语言进行程序设计的方法，以及上机编写C程序的一般步骤。第2章对C语言中使用的数据类型进行了总体介绍，重点讲解了基本数据类型的特点和使用方法，讲述了变量和常量在程序中的作用和输入输出方法。第3章介绍了常用的算法和算法的描述方式，给出了程序设计中常用的三种基本结构，并通过实例讲解了自顶向下、逐步求精的分析过程。第4章讲述了选择结构的C语言实现方法，包括条件判断的描述，if语句、switch语句和条件运算符的使用等。第5章讲述了循环结构的C语言实现方法，以“发现循环—找出构成循环的要素—写出循环语句”为主线，详细介绍了while循环、for循环和do-while循环的相关语法和应用，以及嵌套循环的使用方法。

第6章以函数为主题，描述了对程序进行函数分解的一般原则，总结出根据问题描述编写函数定义的一般原则，并通过实例分析了函数定义、函数声明和函数调用三者间的关系和函数调用机制，最后阐述了变量的作用域和生存期的相关知识。第7章介绍了作为复合数据类型的数组的相关概念和应用，包括数组变量的定义、初始化和相关操作，以及如何通过函数处理数组类型的数据。第8章介绍了另外一种重要的复合数据类型—结构，描述了结构类型的定义、变量的声明和使用，并讲解了结构数组、结构指针以及通过函数处理结构类型数据的方法。第9章重点讲解了指针在C语言中的主要应用领域，并对动态存储管理和链式结构进行了介绍。第10章介绍了文件相关概念，并通过实例介绍了使用标准库函数对文件进行读取操作的方法。

感谢我的家人、同事和中国农业大学出版社的工作人员在本书出版过程中的支持和帮助。本书中的一些观点和提法参考了国内外的优秀书籍和编程爱好者的经验总结，希望能够和广大同行和读者共同讨论研究。由于水平有限，难免存在错误和不足之处，诚心欢迎读者提出批评和意见(作者邮箱：zhoubaishun@126.com)。谢谢关注本书的所有读者。

周百顺

2014年4月于中国劳动关系学院

目　录

第1章　C语言概述

随着网络的发展和计算机软硬件的普及，计算机已经成为家庭和办公的重要工具。计算机不会自动工作，它是由程序控制的，人与计算机打交道的基本方式就是根据自己的需要写出一个程序，而后把这个程序提供给计算机，命令它去执行。此后计算机就会按照程序的规定，一丝不苟地执行其中的指令，直至程序结束。这些用于指挥计算机工作的程序就是使用程序设计语言编写的，C语言是程序设计语言的一种，是程序员与计算机交流的工具。本门课程主要学习如何使用C语言编写程序去指挥计算机工作。

1.1　计算机硬件的组成和工作机制

现代计算机从广泛意义上讲包含硬件和软件两大组成部分，硬件就是我们能够直观看到的构成计算机的各种物理部件，软件则是无形的，软件是用于指挥计算机操作的程序、数据和文档的集合。如果将一台计算机和一个人进行类比，那么硬件相当于人的躯体，是计算机进行工作的物理主体，而软件则相当于人的神经和思想，是计算机的指挥官，指挥硬件去完成指定的功能。我们在商店里看到的计算机通常是指它的硬件组成部分，通常包括机箱和显示器，机箱里面有CPU、内存和硬盘等各种部件。为了更好地理解软件与硬件间的相互作用，首先要了解一下计算机硬件的组成及其工作机制。

1.1.1　计算机硬件的组成

尽管现在市场上出售的计算机在价格、大小、容量和性能上有着很大的差异，但大体上现代计算机都包括如下硬件设备：

- 中央处理单元(CPU)
- 主存储器(内存)
- 辅助存储器(硬盘、光盘、闪存和移动存储等)
- 输入设备(键盘、鼠标、手写板、扫描仪等)
- 输出设备(显示器、打印机、扬声器等)

图1-1是一台现代台式计算机的组成示意图，并通过箭头显示了这些部件在计算机中是如何交互的，箭头的指向代表的是信息的流向。

编写好的程序要想提交给计算机执行，必须要先从辅助存储器中传输到主存储器中。程序执行时通常还需要用户提供一定的初始数据，这些数据可由输入设备提供并存储在计算机的主存储器中。中央处理单元(CPU)通过与主存储器的交互来执行程序指令，访问并处理这些数据，执行结果根据需要可通过输出设备显示出来。下面，我们将从程序运行的角度对硬件组成部分进行更为详细的介绍。

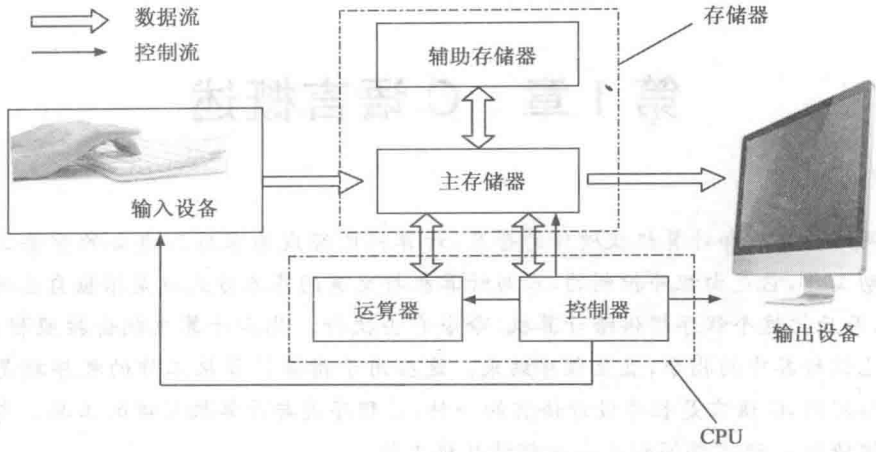


图 1-1　台式计算机的组成示意图

1. CPU

CPU(central processing unit,中央处理器)是计算机的大脑,它进行实际的运算并控制整个计算机的活动。CPU 执行的动作是由程序给出,程序是存储在存储器中一系列计算机能够识别的指令。例如,一条指令可以指示计算机将两个数相加,另一条指令则是将相加后的结果显示在终端屏幕上。每类 CPU 都有自己的指令集,明确了其能够接受并处理的指令的集合。

2. 主存储器

主存储器通常也称内存。计算机中所有程序的运行都是在内存中进行的,当计算机执行一个程序时,必须先将程序本身和运行过程中需要的数据保存到内存中,才能够提供给 CPU 使用。人们将内存系统设计得十分高效,以使得 CPU 能够迅速地读取所需信息。

3. 辅助存储器

尽管计算机运行程序时需要使用内存来保存程序和活动数据,但内存只有在开机通电时才能进行数据的存储,关机后存储在内存中的数据就会丢失。辅助存储器使得计算机能够在电源关闭时永久保存一些数据。当前,计算机最常用的辅助存储器是硬盘,其不仅能够永久保存数据,而且容量通常也远远高于内存容量。在多道程序并行的环境中,内存的容量相比于程序的需要总是不够的,可以借助辅助存储器来实现数据的换入和换出,达到虚拟扩充内存的效果。

4. 输入输出设备

输入输出设备也称 I/O 设备,用于帮助使用者与计算机进行交流,输入设备负责将数据传输给计算机,输出设备则将计算机处理的结果显示给使用者。

1.1.2　二进制与计算机的工作机制

现代计算机大都采用了大规模集成电路,在微小的芯片上集成了数以百万计的晶体管等

元件，通过电路和元件状态的变化来实现计算功能。当计算机处于运行状态，每过来一个电脉冲，这些晶体管的状态就会在导通和断开中变换，从而产生一种新的状态，再来一个电脉冲，状态又变换一次，最终达到目标状态，完成任务。如果用数字 1 代表导通，数字 0 代表断开，则由 0 与 1 构成的二进制串刚好能够表示出电路的所有状态。基于这样的设计，计算机的运算至今仍采用二进制方式。我们平时使用计算机时感觉不到它是在用二进制计算是因为计算机会把你输入的十进制数自动转换成二进制，算出的二进制数再转换成十进制数显示到屏幕上。这种转换对终端用户是透明的。

二进制计算方法符合计算机的物理设计，抗干扰能力强，技术实现和运算规则简单，有利于简化计算机内部结构，提高运算速度，而且易于与其他进制相互转换。二进制数据用 0 和 1 两个数码来表示，它的基数为 2，进位规则是“逢二进一”，借位规则是“借一当二”。图 1-2 给出了二进制的 1101 和 1011 进行加法运算的示意图：

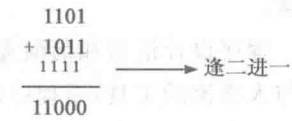


图 1-2　二进制加法运算

二进制数据可以转换为其他进制表示，如十进制、八进制、十六进制等，其中二进制和十进制间的转换最为常用。例如：

$(1011)_2=(1*2^3+0*2^2+1*2^1+1*2^0)_{10}=(8+0+2+1)_{10}=(11)_{10}$

$(89)_{10}=(64+16+8+1)_{10}=(2^6+2^4+2^3+2^0)_{10}=(1011001)_2$

计算机工作时怎么知道自己应该让哪个晶体管导通、哪个不导通呢？这就要靠程序。程序员依据计算机每次要执行的动作事先编好程序，计算机只需按程序进行工作就是了。二进制编码是计算机能够直接识别的唯一语言形式，是最早出现的编程语言，也称为机器语言或低级语言。采用二进制编码编写程序时，编程人员必须记住每个代码的意义，编程难度可想而知。历经多次变革和发展，现在人们已经可以使用更容易为人类理解和使用的高级语言编写程序了。高级语言近似于自然语言，例如，可以使用 BEGIN 表示一段操作的开始，END 表示结束一段操作。高级语言编写的程序并不能被计算机直接识别，为此，人们为每种高级语言都编写了特定的翻译程序，将由 BEGIN、END 等单词构成的高级语言语句翻译成二进制代码，然后交由计算机执行。

1.2　程序设计语言与计算机软件

程序一词源于生活，通常指完成某项事务所需要的一套既定活动方式或活动过程。生活中有很多关于程序的实例，例如，学生去食堂买饭的活动过程可描述如下：

(1)进入食堂；

(2)浏览各个售卖窗口，并告知食堂工作人员想要购买的菜品；

(3)食堂工作人员将相应的菜品盛放在餐盘里；

(4)刷卡支付餐费，取走菜品。

在计算机领域，程序就是使用程序设计语言编写的计算机指令的集合，用于指挥电子计算

机执行一个或多个操作。

1.2.1 程序设计语言概述

语言一词通常指人生活中使用的自然语言，如汉语、英语等。这些语言是人类信息交流的工具和媒介，其描述的程序是为了给人看，要人做的。为了与计算机进行交流，指挥计算机工作，需要一种意义清晰、人用起来方便、计算机也能够处理的描述方式，实现人和计算机之间的交流。这种供人编写计算机能够处理的程序用的语言就是程序设计语言，也常被称为编程语言。

程序设计语言和前面提到的自然语言的区别主要在于交流的对象不同，英语和汉语都是人与人交流的工具，而程序设计语言则是人与计算机交流的工具，不仅人能够懂得和掌握它，使用它描述所需的计算过程，而且计算机也能够“理解”它，可以按照程序设计语言给出的计算过程的描述去执行任务，完成人们所需要的工作。每门语言都有沟通符号、表达方式与处理规则，一般人都必须通过学习才能获得语言能力，由单个词语到整个句子，再到对整个任务的描述。

程序设计语言的发展至少经历了机器语言、汇编语言和高级语言三个阶段。

计算机是人类发明的一种自动机器，能够执行一组基本操作，每个操作能完成一件很简单的工作，如做一次加减乘除运算，或者比较两个数值的大小等。在计算机内部，一切信息都是以二进制编码的形式存在，为了使人们能够指挥计算机工作，每种计算机都提供了一套指令，以及描述其指令的二进制编码形式。每条指令都对应于计算机能够执行的一个基本动作，而所谓的计算机程序就是一组这种指令形成的序列。

这种二进制的描述形式称为机器语言，计算机能够直接“理解”和执行它。用机器语言编写的程序称为机器语言程序。计算机诞生之初，人们只能直接使用机器语言来编写程序。但是，从使用的角度，二进制形式的机器语言很不方便，用它写程序难度很大，工作效率极低，写出的程序也晦涩难懂。下面给出一小段算术运算的二进制代码示例：

```
00000001000000001000
00000001000100001010
00000101000000000001
00000001000100001100
00000100000000000001
00000010000000001110
```

这段二进制代码描述了表达式a＊b＋c的求值过程。一个复杂的程序里可能会有成千上万条类似的指令，执行流程错综复杂，采用二进制编码来完成这样的复杂程序几乎是人力所不能及的事情。

为了解决上述问题，人们很快就设计出采用符号形式的汇编语言。汇编语言采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，从而降低了程序的编写难度，增强了程序的可读性。表达式a＊b＋c的求值过程用汇编语言可描述为：

```
load 0 a
load 1 b
mult 0 1
load 1 c
add 0 1
save 0 d
```

使用汇编语言编写的程序，机器不能直接识别，还要由专门软件(汇编语言编译器)转换成机器指令才能送给计算机去执行。

汇编语言的每一条指令都对应于一条机器指令，但采用了助记符表示，这些符号非常接近于自然语言的要素，使得每条指令的意义更容易理解和掌握，程序设计也就容易多了。但是，汇编语言缺少高层结构的描述，难于从汇编语言代码上理解程序设计意图，可维护性差，面对复杂问题时编程难度大。

为了使计算机能够更方便地为更多人使用，出现了面向用户的高级语言(相比于面向机器的低级语言)。1954 年诞生了第一个高级程序设计语言 FORTRAN，宣告了程序设计的一个新时代的开始。BASIC 语言、PASCAL 语言、C 语言、C＋＋语言、DELPHI 语言、VISUAL BASIC 语言、JAVA 语言和 PYTHON 语言等都是这一时代的杰出代表。

高级语言与计算机的硬件结构及指令系统无关，它更接近于自然语言和数学表达式，通用性较好。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令，具有更强的表达能力，易学易用。高级语言的使用，大大提高了程序编写的效率和程序的可读性，使得有更多人能够并乐于加入到程序设计活动中。高级程序设计语言的诞生和发展，对于计算机的普及和发展起了极其重要的作用。上面的求值过程在高级语言中可直接使用其对应的算术表达式进行描述：

```
a * b＋c;
```

与汇编语言一样，计算机无法直接识别和执行高级语言，必须翻译成等价的机器语言程序才能执行。人们在设计好一个高级语言后，还需要开发与之相配套的实现软件，以便将高级语言编写的代码翻译或解释成计算机能够直接识别的二进制代码。高级语言的基本实现方式有两种—编译实现和解释实现。

1. 编译实现的高级语言

采用编译方式实现的高级语言都有一个对应的翻译软件，用于将该种高级语言编写的计算机程序翻译成基于一定的计算机硬件平台的等价机器语言程序。这种实现方式的翻译过程与执行过程是分开的，首先统一将高级语言编写的程序翻译成机器语言程序(二进制代码)，再通过执行机器语言程序来实现功能。当需要再次执行该程序时，直接执行前面已经翻译好的机器语言程序即可，无须再次翻译。

2. 解释实现的高级语言

采用解释方式实现的高级语言需要有一个对应的解释软件，用于读入这种高级语言编写的程序，并能一步步按照程序的要求一边翻译一边执行，完成程序所描述的工作。有了这种解

释软件，我们只要把写好的程序送给运行着这个软件的计算机，就可以完成相应的程序功能。解释方式实现的过程中，不产生独立的机器语言程序，解释和执行是一体完成的。当需要再次执行该程序时，需要将高级语言编写的程序再次交给解释软件解释执行一次。

当前的实际计算机系统中，两种实现方式都较为常见。如C语言就是采用编译实现的高级语言，PERL、SHELL等语言则属于解释实现的高级语言，也有些语言结合了两种实现方式的优点，采用介于两者之间的实现方式。

1.2.2 计算机软件

未安装任何软件系统的计算机被称之为裸机，只有安装了软件的计算机才能够被普通人所使用。计算机软件也称软件，是指计算机系统中的程序、数据和相关文档的集合。程序是对计算任务的处理对象和处理规则的描述，数据是程序加工的对象，文档是为了方便了解程序所编写的阐明性资料。软件是用户与硬件之间的接口，用户主要是通过软件完成对计算机硬件功能的使用。计算机软件可分为系统软件和应用软件两大类。

1. 系统软件

系统软件通常可细分为操作系统、语言处理系统、服务程序和数据库管理系统四大类。

- 操作系统(operating system，简写为OS)：是管理、控制和监督计算机软、硬件资源协调运行的程序系统，它是直接运行在计算机硬件上的、最基本的系统软件，是系统软件的核心。操作系统的出现使得普通用户能够方便地使用计算机，它能够统一管理计算机系统的全部资源，合理组织计算机的工作流程，以便充分、合理地发挥计算机的效率。常见的操作系统软件包括：DOS操作系统、微软公司的视窗操作系统(Windows系列)、UNIX和Linux操作系统、苹果计算机专用的MAC操作系统等。

- 语言处理系统：如前所述，机器语言是计算机唯一能直接识别和执行的程序语言，如果要在计算机上运行高级语言程序就必须配备程序语言的处理程序，来完成高级语言和机器语言之间的转换。当前，很多语言处理系统将程序的编写功能、编译功能、调试功能等集成到一起，以方便程序员使用。

- 服务程序：能够提供一些常用的服务性功能，为用户开发程序和使用计算机提供方便。如计算机上经常使用的诊断程序、调试程序、编辑程序均属此类。

- 数据库管理系统(data base management system，DBMS)：用于管理数据的计算机软件，主要是针对大数据量情况下的数据存储、检索、查找等应用需求。常用的数据库管理系统包括：Oracle公司的数据库管理系统软件、IBM公司的DB2以及微软公司开发的MS SqlServer等。

2. 应用软件

应用软件可细分为通用软件和专用软件。

- 通用软件：这类软件通常是为解决某一类通用问题而设计的。例如：文字处理、表格处理、文稿演示等，典型的有微软公司开发的Office系列，金山公司开发的WPS系列等。通用软件通常在市场上公开售卖，买回来直接安装使用即可。

- 专用软件：有些具有特殊功能和需求的软件是市场上无法买到，因为它对于一般用户来说太特殊了，所以只能组织人力专门开发。

1.3　C 语言的发展历程

C 语言之所以命名为 C，是因为 C 语言源自 Ken Thompson 发明的 B 语言(BCPL 语言)，是对 B 语言的一种改进，而 B 语言之前还有 A 语言(ALGOL 60 语言)，A 语言取名自世界上第一位女程序员 Ada(艾达)。

1963 年，剑桥大学将 ALGOL 60 语言发展成为 CPL(Combined Programming Language)语言。

1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，于是产生了 BCPL(Basic Combined Programming Language)语言。

1970 年，美国贝尔实验室的 Ken Thompson，以 BCPL 语言为基础，设计出很简单且很接近硬件的 B 语言(取 BCPL 的首字母)。并且使用 B 语言编写了第一个 UNIX 操作系统。

1972 年，美国贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。

1978 年由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。Brian W. Kernighian 和 Dennis M. Ritchie 出版了名著《The C Programming Language》，从而使 C 语言成为目前世界上流行最广泛的高级程序设计语言。自此，C 语言被广泛应用，从大型主机到小型微机，也衍生了 C 语言的很多不同版本。

1983 年美国国家标准局 (American National Standards Institute，简称 ANSI)成立了一个委员会，专门来制定 C 语言标准。

1989 年 C 语言标准被批准，被称为 ANSI X3.159—1989 “Programming Language C”。这个版本的 C 语言标准通常被称为 ANSI C。

1990 年，国际标准化组织 ISO(International Organization for Standards)接受了 ANSI C 作为 ISO C 的标准，也称 C90。1994 年，ISO 修订了 C 语言的标准。

1995 年，ISO 对 C90 做了一些修订，即“1995 基准增补 1(ISO/IEC/9899/AMD1:1995)”。

1999 年，ISO 又对 C 语言标准进行修订，在基本保留原来 C 语言特征的基础上，增加了一些功能，命名为 ISO/IEC9899:1999。

2011 年 12 月 8 日，ISO 正式公布 C 语言新的国际标准草案：ISO/IEC 9899:2011，即 C11。

1.4　C 程序简介

在掌握相对复杂的程序设计内容之前，对程序设计有一个直观的认识是很有意义的。在学习之初，我们首先以一个简单程序开始，从整体上介绍和理解程序。在这一阶段，读者暂时不必拘泥于细节，更为详细的内容将会在后面章节逐步介绍。

1.4.1　C 程序示例

C 语言的一个优点就是让我们可以使用类似于日常英语的语言来编写程序，即使不懂得如何编写程序，也能够阅读并理解一些简单的程序。下面给出两个数求和的示例程序：