高等院校计算机任务驱动教改教材

C语言编程思维

（第2版）

陈　萌　鲍淑娣　编著

清华大学出版社

高等院校计算机任务驱动教改教材

C语言编程思维

（第2版）

陈　萌　鲍淑娣　编著

清华大学出版社
北京

内容简介

本书针对程序设计零基础的读者编写，系统地介绍了如何使用C语言进行程序设计工作。全书从第1章回答十个与编程有关的提问开始，首先向读者介绍了“何谓编程”“为何编程”，以及“编程难吗”等一些程序设计初学者常常提出的问题，使读者在开始学习前对与编程相关的一些重要问题有所了解；第2章介绍了如何搭建一个C语言开发环境；第3章用简洁、平实的语言介绍了如何从机器的视角分析、理解问题，并详细地介绍流程图、伪代码两种编程辅助工具；第4～11章分别详述了基本程序流程控制结构、数组、函数、指针、文件、自定义结构数据类型等C语言程序设计的基础知识。为了帮助读者对例题的理解，全书所有例题有分析、源码和解释部分，其中第4～8章的例题全部配有流程图。

本书适合各类希望了解、学习C语言编程知识的人士，尤其适合作为高等院校各专业、高职高专及中职相关专业C语言程序设计及相关课程教材之用。各章大部分习题来自全国及浙江省计算机等级考试的真题，因此本书也非常适合准备参加各类计算机等级考试的学生学习、辅导之用。

图书在版编目(CIP)数据

C语言编程思维/陈萌，鲍淑娣编著. —2版. —北京：清华大学出版社，2019(2019.12重印)
(高等院校计算机任务驱动教改教材)
ISBN 978-7-302-53590-4

Ⅰ. ①C…　Ⅱ. ①陈…　②鲍…　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(2019)第180550号

责任编辑：张龙卿
封面设计：范春燕
责任校对：袁　芳
责任印制：杨　艳

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
课件下载：http://www.tup.com.cn，010-62770175-4278
印 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：21.75　　**字　　数**：524千字
版　　次：2014年1月第1版　2019年9月第2版　　**印　　次**：2019年12月第2次印刷
定　　价：59.00元

产品编号：084179-01

前 言

编者走进精彩纷呈的计算机世界至今正好20年，从来没有想过要写一本C语言程序设计教程，因为从我学习C语言起，再到后来教授C语言课程，一直都在使用谭浩强先生的C语言教材。那不仅仅是一本经典的C语言教材，更是对自己大学时代的一种记忆与怀念。然而，自从为非计算机专业的学生讲授C语言课程以后，慢慢发现，一些经典的计算机专业教材由于讲授内容较全面、深入，反而不一定很适合他们。作为公共计算机课程的一种，面向非计算机专业开设的C语言程序设计课程，不需要追求掌握了多少语法知识，掌握了多少编程技巧，而应该更多地关注是否通过一门语言工具，使学生们了解计算机程序的运行原理，以及是否掌握了一定的逻辑思维能力，能否以计算机的思维方式去考虑、分析实际问题。即通过这门课程的学习，使各专业的大学生具备基本的计算思维能力，本书正是基于这样的指导思想而写作的。

阅读本书时请注意以下问题。

首先，我希望读者能够对书中加粗、加点的文字内容引起足够的注意，那往往是一些容易被忽视、引发错误的内容。

其次，书中每一个例题在示例代码之前都有分析，之后都有解释，这两处包含了编写程序的一些思想分析和总结，其中不乏一些编程经验和技巧，希望读者不要仅仅将注意力集中在源代码的阅读上，更应该对例题的这两个部分进行细致的阅读。

再次，本书虽然为每一个例题都提供了完整的源代码，却不准备以任何形式向读者提供这些源代码的电子版。实际上，在互联网高度发达的今天，要做源程序的发布非常容易。但是，本人坚持认为，作为一名程序设计的初学者，将每一个例题的源代码自己输入到计算机中本身就是一项重要的练习。你会发现，刚开始的时候，即使对照书本小心地输入，在编译时还是会出现很多错误警告；而读者是在排除这些错误的过程中实现了编程能力的提高。

最后，本书第2版虽然提供了每一章练习题的参考答案，但是仍然不建议读者使用这些参考答案，因为对于看程序写出结果这类习题，读者只需输入题目中的源代码并运行程序就能获得正确的答案。自己运行程序，还可以练习使用断点等方式观察程序运行时变量的变化过程，可以更深入地了解程序的运行。对于编程题，参考答案反而会限制读者计算思想的培养和

形成。正所谓“兵无常势，水无常形”，实现相同功能的源程序也可以多种多样，发散性的思维对于学生尤为重要。请记住，只要能让你编写的程序顺利运行，并输出期望的结果，那它就是答案！当然，如果读者对于部分习题的解决的确毫无头绪时，也可以登录清华大学出版社的官方网站下载本书提供的参考答案。如果读者有任何好的意见、建议或求助，我将十分乐意通过邮件（nbchen75@sina.com）随时为你提供帮助！

本书第 2 版得到了宁波工程学院电子与信息工程学院各位领导及同事的大力协助，特别是在本书修订过程中，理学院的陈明、杨帆、林勇，机械学院的袁云龙、王明军等老师提出了大量宝贵意见，在此向帮助过我的各位同仁表示衷心的感谢！

由于编者水平有限，书中一定存在着各种疏漏与不足之处，恳请各位专家、读者批评、指正，谢谢！

编　者

2019 年 5 月

目　录

第1章 编程十问

长期以来，人们一直梦想着能够发明一种机器，用来帮助人类完成各种繁复的计算工作。数千年来，有的人不断梦想着，有的人不断研究着，直到1614年，苏格兰人约翰·纳皮尔(1550—1617年)发表了一篇论文，其中提到他发明了一种可以计算四则运算和方根运算的精巧装置——纳皮尔的骨头计算器，这种计算器可以说是机械计算器滑尺的直接祖先。现在很多人认为，纳皮尔骨头计算器的发明开启了人类用机器计算的时代。

根据美国联邦法院1973年做出的一项裁决，人类真正意义上的第一台电子数字计算机是1935—1939年间由美国爱荷华州立大学物理系副教授约翰·文森特·阿塔那索夫(John Vincent Atanasoff)和他的研究生克利福特·贝瑞(Clifford Berry)研制成功的，取名为ABC(Atanasoff-Berry Computer)。

而第一台能够编程的电子数字计算机是1946年2月在美国宾夕法尼亚大学摩尔学院的约翰·埃克特和约翰·莫奇利指导下完成的ENIAC(中文名：埃尼阿克)，如图1-1所示。

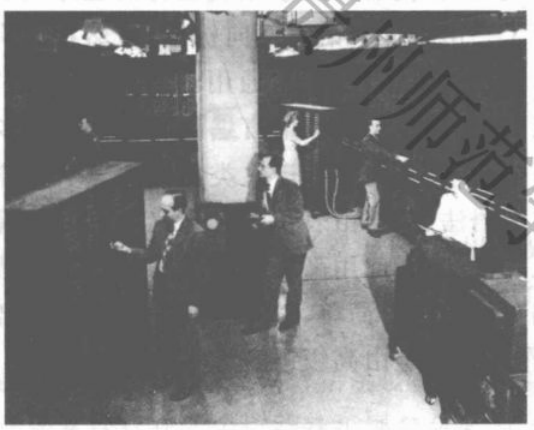

图1-1 第一台电子数字计算机ENIAC

以现在的眼光来看，ENIAC的功能实在无法值得一提，它仅能每秒执行5000次加法或400次乘法，而制造的成本却相当高昂，它使用了18000个电子管、70000个电阻、10000个电容、1500个继电器、6000多个开关，造价高达48万美元(1946年的48万美元)。ENIAC体积巨大，长30.48米、宽1米、占地面积170平方米，重30吨，耗电量150千瓦。

尽管ENIAC的计算速度和功能还不如现在一个最普通的计算器，但这并不妨碍它成为一台划时代的计算机，因为从ENIAC开始，人类正式进入了程序时代。

1.1 何谓编程

现代计算机是通过执行预先编制的不同指令实现工作的，若干条指令的序列即构成了计算机程序。所谓编程，即编写计算机程序的通俗简称，它是指为解决某个问题而使用一种或多种程序设计语言编写程序代码，并通过编译、执行该程序代码后得到结果的过程。编程是一种典型的智力活动，由于计算机只能够理解 0、1 二进制组成的机器代码，无法理解人类的自然语言，为了使计算机能够理解人的意图，人们就必须将需解决问题的思路、方法和手段转换成计算机能够理解的形式，使得计算机能够根据人的指令一步一步地去工作，完成某种特定的任务。为了实现将人的意图转换成计算机所能理解的命令的过程，需要使用到编程语言。编程语言是沟通人类与计算机世界的一座桥梁，它通过在人类与计算机之间进行一系列约定与规则的方式，使人类实现了与计算机之间的便捷沟通。

正如人类有不同语言一样，计算机世界的编程语言也有很多。据一项统计称，电子数字计算机发明至今的 50 余年里，已出现的编程语言种类多达 2500 余个，当然，其中很多编程语言存在着千丝万缕的亲缘关系，例如源自 Bascic 语言的编程语言就有多达 128 种之多。

编程语言按照与计算机硬件联系的紧密程度，一般分为低级语言和高级语言两大类。低级语言是一种在编程时无法进行进一步的抽象化，而与中央处理器的机器语言或指令直接对应的一种编程语言，例如汇编语言即是一种典型的低级语言。低级语言面向机器，与具体机器的指令系统密切相关，一种机器上能够正确运行的程序，换到其他品牌、型号的计算机系统上可能就无法运行了。由于低级语言面向机器，虽然在所有编程语言中具有运行速度快、代码效率高的优点，但是也有编程困难、通用性差、开发效率低的不足。因此，现在除开发一些硬件驱动程序以外，直接使用低级语言编程的场合并不多。

高级语言是一类在语法、指令上更接近于人类自然语言和数学公式的程序设计语言的统称。高级语言按照一定的语法规则，由表达各种意义的运算对象和运算方法构成。由于高级语言是独立于计算机硬件的，因此用高级语言编写的计算机程序通用性好具有较好的移植性。使用高级语言编写程序相对简单、直观，易理解，不容易出错。因此，现在绝大多数的应用程序，甚至一些操作系统都是由高级语言编写而成的。

1.2 为何编程

很多非计算机专业的大学生常常会对为什么要学习编程这个问题感到疑惑，编程不是计算机专业学生才应该学习的知识吗？为什么非计算机专业的大学生还需要学习编程呢？

今天，计算机早已走下了神坛，不再是只有少数人或团体才能够使用的昂贵科学设备，经过几十年的发展，计算机已经成为绝大多数专业不可或缺的重要基础性工具。因此，非计算机专业大学生，特别是理工科大学生，通过学习编程更加深入地掌握计算机工具的运行特点和使用技能，将会对自己本专业的学习打下良好的基础。

另外，学习编程能够培养大学生的逻辑思维能力。培养逻辑思维能力是理工类各专业大学生重要的教学目标之一，由于计算机程序具有极强的逻辑性和抽象性，在编程学习过程中，能够非常有效地训练和培养学习者的逻辑思维能力和抽象思维能力。计算机并不能理解人类思考问题的方法，在编写一段程序时，不可能对任何一个解题步骤进行跳跃式思考，每一个解题步骤都必须严格地按照逻辑规律一步一步地完成，经过一段时间的编程训练，就能够克服人类思维中的模糊性、跳跃性，逐渐养成严密的逻辑思维能力。

学习编程还能够更好地培养理工科大学生的基本工程素养。刚刚走进大学的学生们普遍缺乏基本工程素养，很多学生也根本不了解任何有关工程的知识，这对于即将进入专业领域学习的理工科学生而言是非常不利的。而融入了软件工程思想的现代编程教学能够很好地向初学者介绍一些有关工程的基本概念和思维，使学生在具备了初步的工程素养之后能够更有效地进行专业学习。通过编程培养理工科学生的基本工程素养并非像听起来那样不靠谱，例如，编程学习中的重要知识点函数就非常适合培养和训练学生的工程思维。在任何专业领域，一项复杂的、大型的工程任务通常都需要将其分解成若干个简单的、小型的任务，通过解决这些分割后的小任务，实现大型工程任务的解决。而在计算机程序设计中，函数正是扮演这样一个分解任务的角色，每一个函数只完成任务中的一小部分，通过多个函数的顺序调用，实现一个复杂问题的求解。因此，学习编程知识对于理工科学生培养自己的基本工程素养是十分有帮助的。

"Everybody in this country should learn how to program a computer... because it teaches you how to think." —— Steve Jobs

"这个国家的每个人都应该学习如何写计算机程序……因为它教你如何思考。"——史蒂夫·乔布斯

1.3　怎样编程

由于高级语言更接近人类的自然语言，而非机器语言，所以计算机无法直接理解高级语言编写的程序。使用高级语言编写并最终生成一个计算机能够执行的程序的完整过程一般包括下面几个步骤。

- 编写源程序。源程序是按照某一高级语言语法、符号规则编写的，供人类阅读、交流的文本代码，源程序通常保存在一个以特殊扩展名命名的文本文件中。编写源程序可以使用任何一款文本编辑器，如 Windows 系统自带的记事本、NotePad++、Vim、Emacs 和 TextMate 等。这些第三方文本编辑器大多提供了一些各具特色的文本编辑功能，同时很多也支持语法高亮显示，但是，由于无法实现直接编译、调试源程序，所以选择高级语言编译系统自带的 IDE(Integrated Development Environment，集成开发环境)编写源程序将会更加方便。
- 编译。源程序的阅读对象是人，而非计算机，为了使计算机能够理解源程序的意义，必须对源程序进行编译，即将源程序翻译成为可供计算机阅读、理解的目标程序，完成这一翻译工作的程序即为编译器。编译器能够根据所运行计算机的不同，将源程序翻译成供不同计算机系统阅读的目标程序。需要提醒的是，并非所有源程序都是

以编译方式运行的，不需编译而直接执行源程序的方式被称为解释方式，如 Basic、Java 等语言即采用解释方式运行源程序。

- 连接。现代程序在开发过程中，程序员除自己编写的语句、命令之外，可能还需要调用其他一些库程序、子函数、相关资源等，也可能一个大型程序由多人共同开发，每个人编写程序的一部分，那么这些组成程序的不同部分就需要在最后生成程序时根据需要和次序连接、拼装在一起，这个过程就称为连接。

编程是一项非常复杂的脑力劳动，随着程序规模的逐渐增大，编程的难度也逐渐增加，出现错误的概率更是成倍增长，所以，编程通常无法一蹴而就；相反，需要反复地调试、修改、试运行，再调试、再修改、再试运行，循环往复，直到最终完成程序的编写。

1.4 计算机的世界有何不同

在没有接通电源之前，计算机只是一堆冷冰冰的电子元件。从宏观角度来看，组成微型计算机的通常部件包括机箱、电源、主板、内存条、CPU、硬盘、光驱、显示器、键盘和鼠标等。这些部件构成了微型计算机系统的硬件部分，然而，仅有硬件部分的计算机系统是不完整、无法工作的。与有形实体的计算机硬件系统相对应的计算机系统的另一个重要组成部分——软件系统却是抽象的、无形的实体。尽管软件系统看不见、摸不着，但是，软件系统的存在却是实实在在的，一台缺少软件系统的计算机即使接通电源，也是无法工作的废铁一堆。正因为软件系统对计算机系统的重要意义，有人将软件系统称为计算机系统的灵魂。

出于简化硬件设计的考虑，计算机系统的软、硬件都是使用二进制数据进行工作的，因此计算机世界也被称为“0/1 的世界”。在计算机中，所有的数据最终都将以二进制的形式进行表达、处理和存储，在进入计算机世界之前，请一定先理解这一点。

将人类世界的各种数据与计算机世界的 0/1 数据进行相互转换，也许并不像看起来那么简单。其中最困难的并不是数据转换本身，而是如何正确理解保存在计算机中的二进制数据到底应该对应为人类世界中的哪一类数据。为了进行复杂的信息、情感等交流，人类经常使用的数据花样繁多，有图像、声音、文字、数字等，而计算机世界的数据只有一种——二进制。将各不相同的数据转换成单一的二进制数据并不困难，但是，存在于计算机中的二进制数据却又对应人类中的哪一种数据呢？这就非常容易造成误解。例如，人类世界的字符'0'，将其保存在计算机中时，可转换成八位二进制数据 00110000，但是数字 48 在计算机中对应的二进制数据也是 00110000，当程序在计算机中读出二进制数据 00110000 时，应该将其理解为'0'呢还是 48？再如，保存在内存中的一段图像，其二进制数据看起来却与一段音乐数据别无二致，计算机又是如何判断哪些数据是图像，哪些数据是音乐呢？这个问题是很多编程初学者常常感到迷惑的地方。其实，计算机并不清楚保存在内存、硬盘、U 盘中的数据到底是什么数据，一切对数据的理解和处理都是由程序完成的，而程序又是由程序员，即人编写的，因此，归根结底计算机中的数据应该如何理解和处理是由人说了算的。内存中的一段数据，如果将其送入到显示器中，那它将显示出一幅图像，而如果将其送入到声卡中，那么这段数据就将以声音的形式表示出来。当 C 语言程序员调用函数 putc()处理二进制数据 00110000 时，计算机屏幕中将显示出一个字符'0'，此时 00110000 表示的就是字符'0'；如果程

序员在一个减法表达式中用到 00110000 时，那么它表示的就将是 48。

计算机的世界看起来也许与我们熟知的人类世界有很大的区别，但是，其实它更单纯、更简捷，没有任何的拐弯抹角，一切只唯命令是从，程序员发出什么指令，计算机便忠实地执行什么指令。程序员只要将自己需要完成的工作以正确的命令形式传达给计算机，就能够得到它的响应。

1.5　程序是怎样工作的

要搞清楚计算机程序是怎样工作的，就不能不先提到美国科学家冯·诺依曼。20 世纪初期，随着科学技术在各个领域不断取得进步和突破，人们开始认真地研究起计算机器来，对于这种谁也没有见过的东西，大家争论的焦点在于，制造可以进行数值计算的机器应该采用什么样的体系结构。由于人们被十进制这个人类习惯的计数方法所困扰，所以，当时很多人将研究的重点放在了研制模拟计算机上。20 世纪 30 年代中期，美国科学家冯·诺依曼大胆提出：抛弃十进制，采用二进制作为数字计算机的数制基础。1945 年，他还提出预先编制计算程序，然后由机器按照事先编制的计算程序来执行数值计算工作，即“存储程序”概念。

冯·诺依曼设计思想可以简要地概括为以下三点。

(1) 计算机应包括运算器、存储器、控制器、输入设备和输出设备五大基本部件。

(2) 计算机内部应采用二进制来表示指令和数据；每条指令一般具有一个操作码和一个地址码；其中操作码表示运算性质，地址码指出操作数在存储器中的地址。

(3) 采用存储程序方式。将预先编制好的程序送入到存储器中，然后启动计算机工作，计算机不需要操作人员干预，能自动、逐条取出指令，并执行指令，返回结果。

冯·诺依曼设计思想中最重要之处在于明确地提出了“存储程序”的概念，为了纪念他提出的这一概念，采用该原理的计算机体系结构被称为“冯·诺依曼”体系结构。目前，绝大多数计算机的体系结构仍属于“冯·诺依曼”体系结构。

采用冯·诺依曼体系结构设计的计算机系统，能够自动执行预先存储在存储器中的程序。一段完整的程序是由若干条指令按照一定的次序构成的，因此，程序也被称为“指令序列”。指令是计算机系统中控制器能够执行的一次动作，计算机系统通过连接执行指令实现程序的运行。当程序运行时，组成程序的若干条指令按照预先设定好的次序依次完成以下操作。

(1) 取出指令。从存储器某个存储单元中取出即将要执行的指令，并送到 CPU 内部的指令寄存器暂存。

(2) 分析指令。把保存在指令寄存器中的指令送到指令译码器，由译码器分析产生出该指令对应的操作控制信号，并送往各个执行部件。

(3) 执行指令。计算机中各部件接收到指令译码器发出的相应控制信号，立即完成指令规定的操作，各部件全部完成后，该指令即执行完成。

(4) 准备下一条指令。一条指令完成后，取指机构将自动形成下一条指令的地址，为执行下一条指令做好准备。

计算机程序正是依照上述四个步骤一条一条自动执行完成的。

1.6 为什么选择 C 语言

TIOBE 世界编程语言排行榜展现了编程语言的流行趋势，每一个月都有最新的数据被更新。这份排行榜的数据取样来源于互联网上富有经验的程序员、商业软件开发公司、著名的搜索引擎（诸如谷歌、Bing、雅虎、百度等）的关键字排名，以及 Wikipedia、Amazon、YouTube 统计出排名数据等。当然，这个排行榜只是客观地反映了某个编程语言的热门程度，并不能说明一门编程语言好不好，或者一门语言所编写的代码数量多少。表 1-1 显示的是本书编写之时，2013 年 7 月 TIOBE 的世界编程语言最新排名数据，C 语言毫无争议地再次蝉联第一。如果大家查阅一下 TIOBE 的历史记录就会发现，C 语言多年来一直保持在前两位。

表 1-1 TIOBE 排名图

Position Jul 2013	Position Jul 2012	Delta in Position	Programming Language	Ratings Jul 2013	Delta Jul 2012	Status
1	1	=	C	17.628%	−0.70%	A
2	2	=	Java	15.906%	−0.18%	A
3	3	=	Objective-C	10.248%	+0.91%	A
4	4	=	C++	8.749%	−0.37%	A
5	7	↑↑	PHP	7.186%	+2.17%	A
6	5	↓	C#	6.212%	−0.46%	A
7	6	↓	Visual Basic	4.336%	−1.36%	A
8	8	=	Python	4.035%	+0.03%	A
9	9	=	Perl	2.148%	+0.10%	A
10	11	↑	JavaScript	1.844%	+0.39%	A
11	10	↓	Ruby	1.582%	−0.19%	A
12	14	↑↑	Transact-SQL	1.568%	+0.61%	A
13	15	↑↑	Visual Basic . NET	1.254%	+0.34%	A
14	19	↑↑↑↑↑	PL/SQL	0.920%	+0.28%	A−
15	13	↓↓	Lisp	0.868%	−0.13%	A
16	16	=	Pascal	0.792%	−0.04%	A
17	12	↓↓↓↓↓	Delphi/Object Pascal	0.691%	−0.47%	B
18	20	↑↑	MATLAB	0.680%	+0.04%	B
19	23	↑↑↑↑	Bash	0.622%	+0.04%	A−
20	25	↑↑↑↑↑	Assembly	0.581%	+0.03%	B

如果再仔细地观察一下排名表中前十位的编程语言，就会发现更令人惊讶的结果。其中排名 2、3、4、6、10 的 Java、Objective-C、C++、C# 和 JavaScript 都与 C 语言有着相当密切的亲缘关系，即这五种编程语言的基本语法都源自 C 语言，排名第 5 位的 PHP 语言，其语

法也大量借鉴 C 语言。从这个结果可以看出，掌握了 C 语言，将对世界上 61.561%的源代码不再陌生。而且，掌握了 C 语言之后，再去学习其他的编程语言，特别是学习 Java、C++、C# 等语言将更加容易。

除此之外，C 语言相对其他编程语言还具有很多不可忽视的优势。

(1) C 语言是一种过程性程序设计语言，它的发展贯穿了计算机软件发展的历程，蕴含了程序设计的基本思想，囊括了程序设计的基本概念。

(2) 从统计数据上来看，目前相当多的设备驱动程序是使用 C 语言编写的，C 语言还被用来开发操作系统，目前最著名、最有影响、应用最广泛的 Windows、Linux 和 UNIX 三大操作系统的大量核心代码都是使用 C 语言编写的。

(3) C 语言相比很多其他语言编写的程序，在实现相同功能时，所用的代码行数更少，并且所开发的程序运行效率却更快，体积更小，因此，不少高质量的木马、病毒、恶意代码、漏洞攻击程序都是使用 C 语言开发的。

(4) C 语言语法精练、简洁，与许多程序设计语言在语法上存在着相似性，Java、C++、C#、JavaScript 等更是与其有直接的亲缘关系。因此，掌握了 C 语言编程基础，再转而学习其他的程序设计语言，将起到事半功倍的效果。

(5) C 语言作为世界上最流行的编程语言已经存在很多年了，多年来，在不同的操作系统和应用领域中，由 C 语言开发的代码数不胜数，为后续开发者提供了大量可以重复利用的现成代码，这不但提高了软件的开发速度，而且还可以节省巨大的开发成本。

(6) C 语言仍然是目前软件开发企业普遍需要员工掌握的语言，因此，很多软件开发企业将是否掌握 C 语言以及掌握到何种程度作为评判一位应聘者基本能力的重要条件。

(7) 绝大多数的嵌入式设备都支持 C 语言开发，从微波炉到手机，从洗衣机到飞行控制系统，很多与人类生产、生活密切相关的设备，其运行的程序都是由 C 语言开发的。

如此众多的理由，你是否也对学习 C 语言动心了呢？

1.7　C 语言从何而来

要了解 C 语言的起源，就不能不先提到两位“黑客”鼻祖——肯·汤普逊和丹尼斯·里奇。在“黑客”世界，想要成为顶尖高手，不熟练掌握 C 语言编程和 UNIX 操作系统是不可能练就的，而 C 语言和 UNIX 操作系统都是肯·汤普逊和丹尼斯·里奇的杰作。

1967 年，丹尼斯·里奇进入美国贝尔实验室工作。贝尔实验室是世界上最富创造力的地方之一，这里是晶体管、激光器、太阳能电池、发光二极管、数字交换机、通信卫星、电子数字计算机、蜂窝移动通信设备、长途电视传送、仿真语言、有声电影、立体声录音，以及通信网等许多重大发明的诞生地。自 1925 年以来，贝尔实验室共获得 25000 多项专利，现在，平均每个工作日获得三项多专利。

1965 年，贝尔实验室加入一项由通用电气和麻省理工学院合作的研究计划，该计划要建立一套多用户、多任务、多层次的 MULTICS 操作系统。但在 1969 年时，由于 MULTICS 研究项目的工作进度缓慢，计划被停了下来。同年，肯·汤普逊为娱乐而编写的一个名为“星际旅行”的游戏程序在老旧的 GE-635 计算机上运行时，反应非常慢，正巧被他发现了一