C语言程序设计基础教程

陈朔鹰　陈　英　编著

C

YUYAN CHENGXU

兵器工业出版社

C语言程序设计基础教程

陈朔鹰　陈　英　编著

兵器工业出版社

(京)新登字 049 号

内 容 简 介

本书以计算机的初学者为对象，以美国国家标准化协会颁布的 C 语言的最新版本 87 ANSI C 为基础，讲授 C 语言程序设计的基本概念、函数和语句的形式定义。编程的基本算法和基本技巧，并通过大量的、多种类型的例题和习题，引导读者循序渐进地熟悉并掌握 C 语言设计的方法和技巧。

书中通过对多种饶有趣味的问题的讨论和求解，使读者在轻松、愉快的气氛中理解和探索程序设计的奥妙，从而达到事半功倍的学习效果。本书在加强基础训练、介绍基本算法的同时，较多地选用了具有趣味性的例题，加强可读性，使读者在轻松自然的学习过程中，掌握程序设计的方法。

本书的内容，覆盖了 1994 年 1 月国家教育委员会考试中心编写的《全国计算机等级考试考试大纲》中的二级考试大纲"C 语言程序设计考试要求"，同时，参考了《北京市非计算机专业学生计算机应用知识和应用能力水平考试大纲》对 C 语言的要求，以及其它部分省市计算机应用知识和应用能力水平考试大纲对 C 语言部分的要求。

本书既可以作为高等院校非计算机专业学生的计算机语言教材，也可以作为高等院校计算机专业本科、专科低年级学生学习计算机语言的入门教材，或科技人员自学 C 语言的自学参考书。

图书在版编目(CIP)数据

C 语言程序设计基础教程/陈朔鹰，陈英编著. —北京：
兵器工业出版社，1994.9
ISBN 7—80038—793—3

Ⅰ.C… Ⅱ.①陈… ②陈… Ⅲ.C 语言—程序设计—教材
Ⅳ.TP312

中国版本图书馆 CIP 数据核字(94)第 11396 号

兵器工业出版社出版
(北京市海淀区车道沟 10 号)
新华书店总店科技发行所发行
各地新华书店经销
三河市新艺印刷厂印刷

开本：787×1092 毫米 16 印张：14.375 字数：346.32 千字
1994 年 9 月第一版 1994 年 9 月第一次印刷
印数：0001—5200 册 定价：10.20 元

前　言

随着计算机技术的发展与普及，计算机已经成为各行各业最基本的工具之一，而且正迅速地进入人类生活的各个领域。C 语言作为目前国际上广泛流行的通用程序设计语言，在计算机的研究和应用中已展现出强大的生命力。C 语言兼顾了诸多高级语言的特点，是一种典型的结构化程序设计语言。它处理能力强，使用灵活方便，应用面广，具有良好的可移植性，既适合于计算机专业人员编写系统软件，又适合于应用开发人员编写应用软件。

本书是作者在多年对计算机类和非计算机类本、专科学生 C 语言教学工作的基础上，根据教学实践和学生的学习情况，编写的针对非计算机专业的计算机应用基础课教材。本书立足于培养学生严谨的、良好的程序设计风格和习惯，以 C 语言作为推广计算机应用的启蒙语言，既满足奠定好程序设计基础的需要，又能有较高且适当的起点，担当起普及和提高的双重任务。为此，集我们数年从事 C 语言教学工作和 C 程序开发应用实际工作的经验，我们认为：对于 C 语言的初学者和科技工作者，不适于也不可能在一本书中展开 C 语言的全部内容，这样会导致内容的冗长和分散，影响读者对 C 语言真正特点的理解，反而导致“欲速则不达”。

本书侧重于 C 语言基本知识，着重讲解概念，深入浅出地讲解编程的基本算法和编程基本技巧。对于初学者来说，掌握程序设计的技术和方法，在最初往往是枯燥乏味的。作为一种尝试，我们在教学中曾经将各种程序设计的技术和方法融于趣味问题之中，通过对多种饶有趣味的问题的讨论和求解，使读者在轻松、愉快的气氛中理解和探索程序设计的奥妙，从而达到事半功倍的学习效果。基于这样指导思想，本书在加强基础训练、介绍基本算法的同时，较多地选用了具有趣味性的例题。其目的就是要加强本书的可读性，使读者在轻松自然的学习过程中，掌握程序设计的方法。

本书以实际应用为目的，删除了其它教科书中对 C 语言中过于“技巧性”问题的讨论与介绍。例如，对于“y＝＋＋x＊＋＋x”和“printf(“%d\n”, ＋＋x＊＋＋x)”到底等于几的讨论。因为这类问题多出现在各种考试中，而在实际的编程过程中，几乎没有人这样编写程序。

本书对 C 语言的介绍以美国国家标准化协会颁布的 C 语言的最新版本 87 ANSI C 为基础，介绍 C 语言的基本概念、函数和语句的形式定义，应用说明及要点，并通过大量的、多种类型的例题和习题，引导读者循序渐进地熟悉并掌握 C 语言程序设计的方法和技巧。

本书介绍的 C 语言，覆盖了 1994 年 1 月国家教育委员会考试中心编写的《全国计算机等级考试考试大纲》中的二级考试大纲“C 语言程序设计考试要求”，同时，参考了《北京市非计算机专业学生计算机应用知识和应用能力水平考试大纲》对 C 语言的要求，以及其它部分省市计算机应用知识和应用能力水平考试大纲对 C 语言部分的要求。

书中全部的例题和习题均在 Turbo C 环境下调试、运行，并已汇集在一张软盘上，与本书可配套发行，以方便读者上机学习。同时，为强化读者对 C 语言的学习、理解和应用，根据教学的要求，还编写了《C 语言趣味程序百例精解》(已由北京理工大学出版社出版)一书，该书可作为教师的教学参考用书和学生学习 C 语言的自学参考书。

本书既可以作为高等院校非计算机专业学生的计算机语言教材，也可以作为高等院校计算机专业本科、专科低年级学生学习计算机语言的入门教材。作者曾经使用本书作为非计算机

专业学生学习第一门计算机语言的教材，收到了良好的效果。本书也可以作为科技人员自学C语言的自学参考书。

由于本书的特点，对于已经具有良好的计算机基础的读者和需要提高C语言编程应用水平的读者，不适合选用本书，因为本书编写的目的更多地是面向初学者。

本书的绪论和第一、二、五、七章由陈英编著，第三、四、六、八章和附录由陈朔鹰编著。在编写本书的过程中，自始至终得到了北京理工大学彭一苇教授的热心指导和帮助，他认真审阅了全部书稿，提出了宝贵的意见；在本书的修改过程中，得到了崔桂华老师的帮助，他根据自己的教学实践和使用本书的体会，对本书提出了宝贵的意见，并认真审阅了全部书稿；在本书的编写过程中得到了北京理工大学教务处各位领导，及北京理工大学曾经讲授过C语言的有关教师的大力帮助，他们对本书的修改提出了很好的建议，在此一并表示衷心的感谢。

鉴于作者水平有限，书中一定存在不少错误和不妥之处，敬请读者批评指正。

编著者

1994年9月

目　录

绪　论

一、 计算机语言的演变及发展

现代科学的迅猛发展使电子计算机几乎进入了人类生活的一切领域。计算机已经成为科学工作者的有力助手。目前人和计算机之间还不能象人与人之间那样，完全用自然语言进行交流，表达意图，交换信息。计算机与人的通信要借助于“计算机语言”。这种语言是用来表达计算机程序的，而程序正是人们思想的体现。因此，程序设计是学习计算机应用，掌握计算机工具的中心活动，计算机语言则是进行程序设计的主要工具。

计算机语言特别是高级程序设计语言的产生和发展，推进了计算机的普及和应用。同时计算机科学的发展史亦紧密联系着程序设计语言的发展史。计算机语言的诞生、发展和不断地更新换代，经历了从低级向高级的发展过程。

计算机问世的初期，人们都是用机器语言来编写程序。机器语言是由计算机的指令系统提供的，完全是0、1组成的二进制信息。用机器语言来编写程序是很烦琐的事情，既浪费时间又容易出错，机器指令程序设计难学、难记、难写、难修改。而且每种型号的计算机都独有一套自己的指令系统，程序不便于交流和推广应用，影响了计算机的普及。

鉴于机器语言本身的这些缺陷，人们发展了汇编语言。所谓汇编语言实际上是用一些助记符来代替机器语言的那些代码，使机器语言符号化。汇编语言比机器语言进了一步，但是仍然保留着机器语言的许多弊病，没有摆脱对具体机器的依赖性。

随着计算机的迅猛发展，亟需解决的是计算机硬件的高速度和程序编制的低效率之间的矛盾，在50年代末期产生了“程序设计语言”(相对于机器指令和汇编语言也称为高级程序设计语言或算法语言)。程序设计语言比较接近自然语言，具有直观性、精确性、通用性等特点，而且易学、易懂。

自第一个高级语言问世以来，已产生过上千种程序设计语言，常使用的也有数百种。如果对这些语言进行分类，可以按语言的应用范围、使用方式、功能等不同角度进行分类。

从语言的应用范围来分，可划分为通用型和专用型语言两大类。

具体划分，又可分为适合于数值计算的语言，如常用的FORTRAN语言、ALGOL—60语言；结构化程序设计语言，如C语言、PASCAL语言；适合于商用和管理领域的语言，如COBOL、FOXBASE等。还有一些交互型的通用语言，如BASIC、APL语言。而专用语言更是种类繁多，功能各异。如适合于数控操作的数控语言APL，适合于计算机辅助设计的AHPL和DDL语言，适合于符号处理的SNOBOL、LISP、COMIT语言，适合于人工智能的PROLOG等。另外，还有综合各类语言特点、功能庞大、适用范围颇广的汇集性语言，如著名的ADA和PL/I语言。

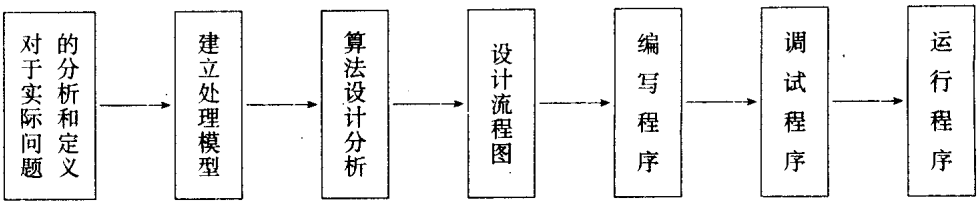
面对这么多的高级语言，要全面掌握几乎是不可能的。事实上，在实际计算机应用中，真正常用的语言不过十几种，像BASIC、FORTRAN、PASCAL、C，还有ADA、COBOL、PL/I等语言。这些语言作为最常用的计算机语言，使用频率和应用覆盖面都是很高的，因此作为初学者，

要把精力集中到这些语言上，特别是应以一种语言为模式，深入学习和应用，掌握该语言的基本组成、结构，编程方法等实现技术，这样才能为学习其它高级语言奠定基础。

二、 程序设计基础

1. 用计算机解决实际问题的步骤

程序设计，简单地讲就是根据要用计算机处理的实际问题，编写相应的能实现该问题处理的计算机程序。程序设计的一般过程可用下面的流程框图表示：



以上程序设计过程的各个步骤一般是有序的，但起始点可以不同，有些阶段可隐去或合并。其中：

① 定义问题　这是程序设计的第一步，也是程序设计的目标。应当通过对实际要处理问题的深入分析，准确地提炼、表达出用计算机要解决的问题，明确要求。

② 建立处理模型　实际问题都是有一定规律的物理过程，用特定的方法描述问题的规律和其中的数值关系，是确定计算机实现算法的理论准备。如，求解图形面积一类的问题，可归结为数值积分，则积分公式即是为解决这类问题建立的数学模型。

③ 算法选择与设计　将要处理的问题分解成计算机能够执行的若干特定操作，也就是确定解决问题的算法。例如，上面提到的用数值积分求解面积的数学模型，不适用于计算机直接求解，因为计算机不能识别积分公式，需要将公式变为等价的计算机能接受的基本运算，如可以选择梯形公式或辛卜生公式来计算，同时还要考虑算法简单，保证计算精度等问题，这些都属于算法选择与设计。

④ 流程图设计　对比较复杂的问题，在编写程序前，形象地表示算法，给出处理步骤的流程图，能直观地反映出所处理的问题中较复杂的关系，使编程序思路清晰，避免出错。流程图是程序设计的良好辅助工具。另外，流程图作为程序设计资料也便于交流。

例如，求 a 和 b 两个数中的最大值。根据此问题的处理要求，可用流程图(见图 0.1)表示程序的处理步骤。

程序流程图是由一些规定的图形、连线及文字说明组成。常用的标准图形和符号如图 0.2 所示。

⑤ 编写程序　用某种高级语言，按流程图描述的步骤写出程序，也称为编码。

⑥ 调试程序与运行程序　将写好的程序上机进行检查、编译和试运行，纠正其中的错误后，正式运行程序。

2. 高级语言程序的翻译和执行

计算机只懂机器语言和机器语言程序，并不能直接执行高级语言编写的程序，这就需要借助“翻译”，将高级语言程序翻译成机器语言程序，然后才能执行。这个“翻译”就是计算机中的系统软件，称为编译程序。用高级语言编写的程序在计算机上调试和运行的实际过程可用下图(见图 0.3)表示。

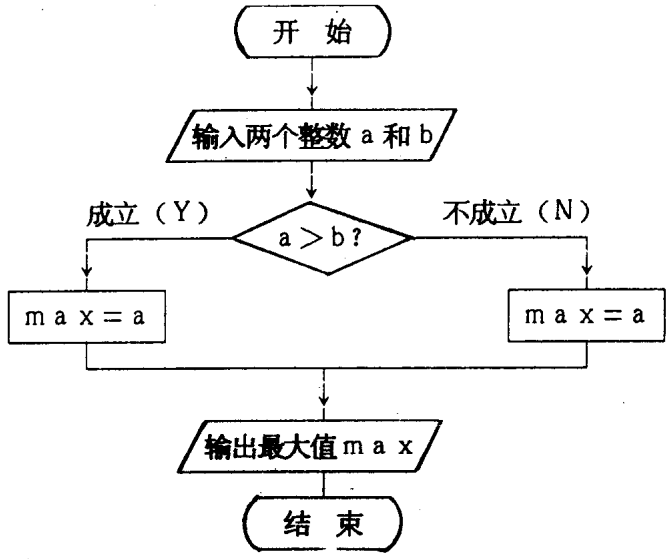


图 0.1　求 a、b 两数中最大值处理流程图

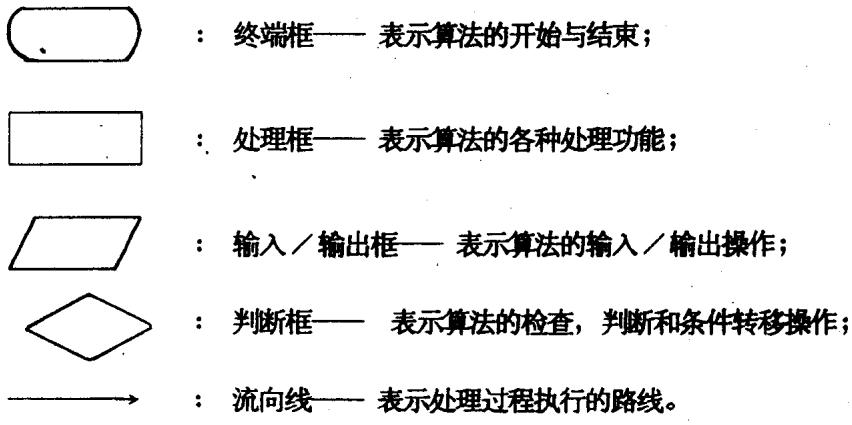


图 0.2　程序流程图形、符号

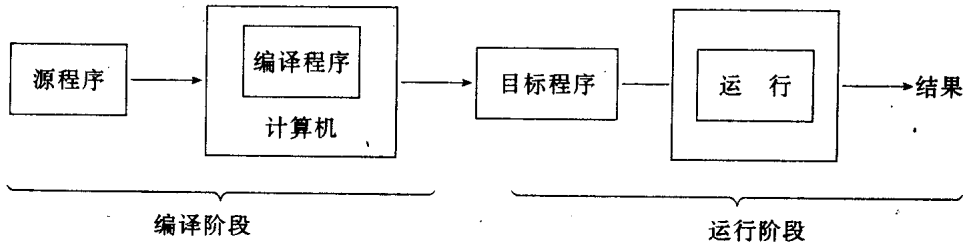


图 0.3　高级语言程序机调试运行过程

整个调试和运行过程分为翻译阶段和运行阶段。图中给出的源程序，是指用高级语言编写的程序，如用 C 语言写的程序称为 C 源程序。目标程序是指编译程序把源程序翻译成等价的机器语言的程序。这些概念对后续学习 C 语言是有用的，至于编译系统本身的实现不必细究，只要知道它的功能即可。

第一章　C语言基本知识

§1.1　C语言简介

1.1.1　C语言的发展及应用

C语言是目前国际上广泛流行的一种通用程序设计语言。它从诞生至今虽只有近二十年的历史，但其发展的速度和应用的范围，却是任何一种程序设计语言所无法比拟的。C语言适应的机种从8位微型机直到巨型机，应用的范围从系统软件到涉及各个领域的应用软件，C语言被作为现代计算机语言的代表之一，展现出强大的生命力。

C语言的研究和诞生起源于系统程序设计的深入研究和发展，它作为书写UNIX操作系统的语言，伴随着UNIX的发展、流行而得到发展与普及。

1967年，英国剑桥大学的理查兹(M. Richards)在CPL(Combined Programming)语言的基础上，实现并推出了BCPL(Basic Combined Programming Language)语言。

1970年，美国贝尔实验室的汤普森(K. Thompson)以BCPL语言为基础，设计了一种类似于BCPL的语言，称为B语言。他用B语言在PDP－7机上实现了第一个实验性的UNIX操作系统。

1972年，贝尔实验室的里奇(D. M. Ritchie)为克服B语言的诸多不足，在B语言的基础上重新设计了一种语言，由于是B的后继，故称为C语言。

1973年，贝尔实验室的汤普森和里奇合作，主要用C语言重新改写了UNIX操作系统，此后，随着UNIX操作系统的发展，C语言的应用越来越多，影响越来越大。

至此以后，C语言不断得到改进，但主要还是作为实验室产品在使用，并依赖于具体的机器。直到1977年才出现了独立于具体机器的C语言编译版本。C语言的独立和推广也推动了UNIX操作系统在各种机器上的迅速实现。

1978年，科尼翰(B. W. Kernighan)和里奇正式出版了《The C Programming Language》一书，此书即成为现在广泛使用的C语言版本基础，亦被称为标准C语言。

1983年，美国国家标准化协会(ANSI)颁布了C语言的新的标准版本“ANSI C”。“ANSI C”比标准C有很大的扩充和发展。

1987年，美国国家标准化协会在综合各种C语言版本的基础上，又推出了美国标准C语言版本，它是目前功能最完善、性能最优良的C语言新版本，称之为“87 ANSI C”。

目前在我国PC系列兼容机上常用的、实际使用频率比较高的C语言版本主要有：Borland International公司的Turbo C(V2.0，V3.0)；Microsoft公司的Microsoft C(V5.0，V6.0，V7.0)和Quick C；Computer Innovasions公司的C 86 (V2. 3) ；Lattice公司的Lattice C (V4.0)等。本书讲述的内容是以“ANSI C”为基础，并参考“87 ANSI C”。

1.1.2 C语言的特点

C语言的强大生命力与其本身具备的突出特点直接相关。从语言体系和结构上讲，它与Pascal、ALGOL 60等语言相类似，是结构化程序设计语言。但从用户应用、实现难易程度、程序设计风格等角度来看，C语言的特点又是多方面的。在此，仅就C语言中被广大软件人员所公认的一些特点归结如下：

1. 语言本身简洁，使用灵活，便于学习和应用。在源程序表示方法上，与其它语言相比，一般功能上等价的语句，C语言的书写形式更为直观、精练。

2. 语言的表达能力强。C语言是面向结构化程序设计的语言，通用直观；运算符达30种，涉及的范围广，功能强。可直接处理字符，访问内存物理地址，进行位操作，可以直接对计算机硬件进行操作，它反映了计算机的自身性能，足以取代汇编语言来编写各种系统软件和应用软件。鉴于C语言兼有高级语言和汇编语言的特点，也可称其为“中级语言”。

3. 数据结构系统化。C语言具有现代化语言的各种数据结构，且具有数据类型的构造能力，因此，便于实现各种复杂的数据结构的运算。

4. 控制流结构化。C语言提供了功能很强的各种流控制语句（如if、while、for、switch等语句），并以函数作为主要结构成分，便于程序模块化，符合现代程序设计风格。

5. 语言生成的运行程序质量高，程序运行效率高。试验表明，C源程序生成的运行程序的效率仅比汇编程序的效率低10%～20%，但C语言编程速度快，程序可读性好，易于调试、修改和移植，这些优点是汇编语言所无法比拟的。

6. 可移植性好。统计资料表明，C编译程序80%以上的代码是公共的，因此稍加修改就能移植到各种不同型号的计算机上。

7. C语言存在的不足之处是：运算符和优先级过多，不便于记忆；语法定义不严格，编程自由度大，编译程序查错纠错能力所限，对不熟练的程序员带来一定困难；C语言的理论研究及标准化工作也有待推进和完善。为此，C语言对程序设计人员的素质要求相对要高。

综上所述，C语言把高级语言的基本结构与低级语言的高效实用性很好地结合起来，不失为一个出色而有效的现代通用程序设计语言。

§1.2 C源程序结构

本节通过两个具体的C语言源程序实例，介绍C语言程序的基本组成和结构，使读者对C语言和C源程序的特性建立感性的了解。

例C1__2001.C：①

```
/* small.C __ The smallest C program. */
main ()
  {
  }
```

① C1__2001.C是与本书配套的软盘上所存的源程序文件名，其中“C1”代表第一章，“2”代表本章的第二节，“001”是本章C源程序实例的序号，“.C”是C源程序文件名的扩展名。

这个C语言源程序可以说是一个最小的C程序，它仅由一个主函数构成，且程序体为空，程序运行时无任何实际的执行动作。本程序的第一行是注释部分，注释内容用一对"/*"和"*/"括起来，注释是为程序设计人员及程序使用者方便理解程序而附加在程序中的说明信息，对程序的编译和运行均不起作用。第二行是C程序的主函数首部，main即为主函数名，表示"主函数"，主函数是一个特殊的函数，每个C程序都必须有一个主函数，它亦是C程序运行的起点，main后的"( )"是函数的参数部分，可为空，但括号不能省。第三行和第四行对应一对花括号"{ }"，表示函数体的开始和结束，"{ }"内语句的集合构成函数体，函数体也可为空。

例C1__2002.C：有人提出一个求素数的公式：

$$p=n*n+n+41$$

并断言一切自然数n，代入此公式后，所求得的p都是素数。现在试采用从n=1开始，依次选择n=2,3,…，将n直接代入公式来验证与n所对应的p是否为素数，若是，则使n值依次递增继续进行验证，若不是，则输出有关信息且终止程序运行。实现此算法的C程序如下：

```
main()                          /* 主函数 */
{                               /* 主函数体开始行 */
  int n,p;                      /* 变量定义、说明 */
  for ( n=1; ; n++)             /* 循环语句,控制n值依次递增,完成相应的验证 */
  { p=n*n+n+41;                 /* 赋值语句,完成p值的计算 */
    printf("%d,p=%d",n,p);      /* 输出变量n和p的值 */
    if (s(p)==0)                /* 条件语句,条件表达式中调用了函数s */
       printf ("ok\n");         /* 输出语句,确认为素数时输出"ok!" */
    else
    { printf ("error.\n");      /* 输出语句,确认p为非素数时输出"error." */
      break;                    /* 间断语句,终止程序执行 */
    }                           /* 此处{...}为一复合语句 */
  }
}                               /* 主函数体结束行 */
s(v)                            /* 函数s(),v为函数s的形式参数表 */
  int v;                        /* 变量说明,v是整型变量且v是函数s的形式参数 */
{ int j;                        /* 变量说明 */
  if (v<2) return (-1);         /* 条件语句,条件表达式成立,执行返回 语句return,返
                                   回非素数标志值 -1 */
  for ( j=2; j<v; ++j)          /* 循环语句,实现判定v是否为素数 */
    if (v%j==0) return(-1);     /* 条件语句是循环的循环体 */
  return (0);                   /* 返回语句,返回确认为素数的标志值0 */
}
```

本程序每个语句的性质及功能由其右部的注释部分给出简要说明。分析此例可知，本程序由两个函数组成，即主函数main()和子函数s()，主函数的具体功能是计算p值，输出验证结果；子函数s的功能是判断一个数是否为素数，其中的返回语句将判定结果的标志值带回到

主函数中，具体是通过函数名 s 带回到主函数的调用处(主函数中条件语句内的 s(p)处)。C 语言的函数是由基本语句、复合语句、函数调用及注释行构成的。

通过以上二个例子，可以对 C 源程序的概貌和特性建立几个基本印象：

1. C 程序是以函数为基本单位，由函数组成。一个完整的 C 程序至少要有一个且仅有一个主函数，它是程序启动时的唯一入口。除主函数外，C 程序还可包含若干其它函数。

2. 函数是由语句和注释组成的。一般函数由两部分组成，一是函数的说明部分，包括函数名、函数类型、形式参数等的定义和说明。另一部分是函数体部分，它是一个函数最外层的一对 { } 括起来的部分。

3. 语句是由一些基本字符和定义符按照 C 语言的语法规定组成的。

4. 每个语句以分号结束。

5. C 程序书写格式比较自由，一行内可以写几个语句，一个语句也可分写在多行内。

§1.3 C 语言的基本组成

任何程序设计语言如同自然语言一样，都具有自己一套特定的、专用的字符集和定义符，C 语言亦如此。这些字符集中的字符和定义符构成了 C 程序的最小的语法单位。

一、 基本字符集

C 语言的基本字符集包括：

数 字：0 1 2 3 4 5 6 7 8 9

字 母：A B C …… Z a b c …… z

(注意：字母的大小写是可区分的。如：abc 与 ABC 是不同的)

运算符：+ − * / % = < > <= >= != == << >> & |

&& || ^ ~ ( ) [ ] { } −> . ! ? : , ;

特殊符号和不可显示字符：_(连字符或下划线) 空格 换行 制表符

书写程序要具有良好的习惯，力求字符确切、工整、清晰，尤其要注意区分一些字形上容易混淆的字符，避免给程序的阅读、录入和调试工作带来不必要的麻烦。

二、 关键字

C 语言有一些具有特定含义的关键字，作为专用的定义符，这些关键字不允许用户作为自定义的标识符使用。C 语言关键字是由小写字母构成的字符序列，它们是：

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	sizeof	static	struct	switch	typedef	union
unsigned	void	volatile	while			

三、 标识符

在 C 语言中，标识符是用户对实体定义的一种定义符，用来标识用户定义的变量名、函数名、文件名、数组名、类型名等。

C 语言中标识符构成规则为：由字母或下划线(连字符)开头的，后面跟由字母或数字或下

划线(或空串)组成的字符序列,一般有效长度是8个字符,ANSI C标准规定的有效长度可达31个字符。下面给出一些合法的标识符及不合法的标识符以具体说明:

合法的C标识	不合法的C标识符(说明)
call__name	call...name(非字母数字或下划线组成的字符序列)
test39	39test(非字母或下划线开头的字符序列)
__string1	-string1(非字母或下划线开头的字符序列)

§1.4 C语言上机一般步骤

编写C语言程序仅是完整的程序设计工作中的一个阶段,还需进行程序的上机调试运行,直至得到正确的运行结果。一般来说,C源程序上机运行要经过以下四个步骤:

1. 在机器上建立C源程序。即用机器上的某种编辑程序或C编译系统本身所具有的编辑环境将C源程序输入计算机,经修改确认无误后,以文件形式存入计算机,称之为C源文件。一般C源文件名的定义为:"文件名.C"。其中文件名是用户对录入且存入文件系统的C源程序指定的名字,用标识符来表示。后缀".C"表示文件的类型是C语言程序。

要注意的是,一个C源程序是由一个或多个函数组成的,一个C源程序可存入一个或几个C源文件中,这样每个源文件可以单独编译。

2. 对C源文件进行编译。这一步工作即是对C源程序进行词法和语法检查,最终翻译成便于机器运行的目标程序(目标文件),目标程序的文件名是:"文件名.obj"。后缀".obj"表示文件类型是目标程序。

3. 连接。将目标程序同必要的库函数或其它目标程序连接在一起,生成可执行程序。可执行程序的文件名为:"文件名.exe"。

4. 执行程序。实际运行经编译、连接生成的可执行程序。如果程序运行结果不正确,可重新回到第一步,重新对程序进行修改、编译和运行。

必须指出,对不同型号计算机上的C语言版本,上机环境各不相同,编译系统支持性能各异,上述步骤有些还可再分解,或集成进行批处理,但逻辑上基本如此。

习 题

1.1 C语言的主要特点是什么?

1.2 C语言程序结构的特点是什么?它由哪些基本部分组成?

1.3 C语言标识符的作用是什么?命名规则是什么?与关键字有何区别?

1.4 指出下列符号中哪些是C语言标识符?哪些是关键字?哪些既非标识符亦非关键字?

stru	au__to	__auto	sizeof	3id	file	m__i__n
-min	call..menu	hello	A B C	SIN90	n*m	/n
x1234	until	cos2x	1234	1234hello	s+3	s__3

1.5 什么是标准C和ANSI C?

1.6 为什么可以称C为"中级语言"?