HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

C 语言程序设计教程

C YU YAN CHENG XU SHEJI JIAO CHENG

秦友淑　曹化工　编著

华中理工大学出版社

C语言程序设计教程

秦友淑　曹化工　编著

华中理工大学出版社

图书在版编目(CIP)数据

TS193/21

C语言程序设计教程/秦友淑　曹化工　编著
武汉:华中理工大学出版社,1996年5月
ISBN 7-5609-1185-4

Ⅰ.C…
Ⅱ.①秦…　②曹…
Ⅲ.计算机语言-C程序设计-教材
Ⅳ.TP36

C语言程序设计教程
秦友淑　曹化工　编著
责任编辑:唐元瑜　王有登

华中理工大学出版社出版发行
(武昌喻家山　邮编430074)
新华书店湖北发行所经销
华中理工大学出版社印刷厂印刷

开本:787×1092　1/16　印张:17　字数:410 000
1996年5月第1版　1999年9月第5次印刷
印数:14 001—19 000
ISBN 7-5609-1185-4/TP·158
定价:18.80元

内容简介

本书遵照C语言标准，结合作者多年教学和科研实践的经验和体会，全面系统、深入浅出地阐述了C语言的基本概念、语法和语义，以及用C语言进行程序设计的基本方法和技巧。内容包括数据类型和表达式、流程控制、函数与程序结构、输入输出与低层接口。尤其对C语言的重点、难点和特色之处，例如表达式、位运算、类型转换、变量的存储类型和指针等，都作了明确、详细和深入的介绍。

本书概念准确，结构合理，层次清晰，实例丰富，选材精心，语言通俗易懂。每章末所附习题可供不同层次和应用的读者选作练习。

本书提供了一本准确而又较全面反映标准C的程序设计教材，既可供高等院校计算机和非计算机专业本科、专科或培训班教学使用，也可供广大科技工作者和研究生自学参考。

阅读和使用本书，不要求读者一定具备高级语言程序设计的基础。

前　言

C语言是一种优秀的通用程序设计语言。由于它的功能强、简洁、灵活和兼有高级语言与低级语言的优点，目标程序效率高，可移植性好，以及它在各种系统上的普遍实现等特点，因此它已成为当今世界上的主流程序设计语言之一。近年来，C语言在我国的实际应用中也已占了统治地位，几乎所有的大专院校都开设有C语言课程，并且有的学校已将C语言程序设计直接作为第一门计算机语言程序设计课程。此外，C语言标准的制定标志着C语言发展的成熟，标准C综合了现代C语言对传统C语言功能的许多重要扩充。C语言创始人和经典著作"The C Programming Language"的作者，按照ANSI C标准改写了他们的原著，于1990年正式发表了该书的第二版"The C Programming Language Second Edition"。当今，几乎所有厂商推出的高版本的C编译程序都是按照标准C实现的(C语言的国际标准和我国国家标准都是以ANSI C为基础制定的)。目前国内已出版的关于C语言的书很多，但准确全面反映标准C的书却很少。另一方面，大多数关于C语言的书或者面向专门应用，例如窗口和图形程序设计；或者要求读者具有一门以上高级语言的基础；或者内容不够深入，不能适应当前的教学需要。

本书正是为适应当前客观情况的需要，以"The C Programming Language Second Edition"为主要参考，结合作者在华中理工大学计算机系多年教学(C语言程序设计、PASCAL语言程序设计、计算机图形学等课程)和科研(C编译程序和PASCAL编译程序等项目)实践的经验和体会，同时参考了国内许多正式或非正式出版的关于C语言的著作而写成的。旨在提供一本准确而又较全面反映标准C的程序设计教材和自学读物。

本书概念准确，结构合理，层次清晰，语言通俗易懂。书中实例丰富、选材精心，在重点突出语法和用法的基础上，注意介绍程序设计方法和一些实用的算法。在内容上，全面、系统、深入浅出地阐述了C语言的基本概念、语法、语义以及用C语言编程的基本方法和技巧。尤其对C语言的重点、难点及其独具特色之处，例如表达式、位运算、类型转换、变量的存储类型和指针等，都作了明确的、详细深入的介绍。每章末所附的各类习题可供不同层次和应用的读者选作练习。本书适于高等院校计算机和非计算机专业本科、专科或培训班用作高级语言程序设计课程的教材，也适于各类科技工作者和研究生自学或参考。

本书用作不同教学对象时，其内容剪裁建议如下：

(1)对于已具有计算机基础知识和一门以上高级语言程序设计基础的各类学员，1.1节和3.3节可以不讲授(但学员应自己阅读，这对后面的学习会有帮助)；第四章可着重讲授语法和用法而不讲实例的算法；第六章可以简要地介绍数组的

说明和引用，而将重点放在对C语言中数组的初始化、数组作函数参数和对多维数组的处理方法的介绍上。

(2)对于非计算机专业本科学员，目录中加“＊＊”的章节可以不讲授。

(3)对于专科或培训班学员，目录中加“＊”和“＊＊”的章节均可不讲授。

本书正文中配有大量实例，所有的实例程序均在turbo c 2.0上调试通过。为了突出重点和节省篇幅，书中有的实例仅给出函数而未列出完整的程序，对于给出完整程序的实例一般都有输出结果。有的实例侧重于说明语法和用法，有的则为了介绍算法或说明应用，讲课时，教师可根据需要选讲部分实例，其余留给学员自己阅读或上机练习。

C语言的语法现象比较复杂，数据的类型转换和表示灵活多变。因此，本书在叙述时采用了循序渐进、逐步深入的方法，对书中出现的每个语法成分，均以非形式化的方法，严格按附录B所给出的C语言语法阐述其概念和规则。

学习和掌握C语言最有效的方法是实践。实践可分为三个层次和两个方面。三个层次是：阅读别人写好的程序(或函数)，理解程序所要完成的任务(即程序功能)，从中学习编程的方法和技巧；模仿编写功能类似的程序；自己独立设计和编写完成指定任务的程序。两个方面是：在条件有限的情况下，动手在纸上严格按语法规则一丝不苟地写出程序；在有条件的情况下应尽量上机练习，调试运行自己写的程序。

此书全部内容的编写及程序的调试均由作者完成。

感谢华中理工大学计算机系领导和同事对此书编写工作所给予的大力支持，感谢华中理工大学出版社的有关同志为此书的编辑和出版所付出的辛勤劳动。还要感谢华中理工大学计算机系学生曹鹏凌同学和机械学院学生曹鹏彬同学在书稿的誊写整理中所做的工作。

热诚地期望此书能为广大读者学好和用好C语言作出一点贡献。由于作者水平有限和时间仓促，书中错误和疏漏之处恳请广大读者批评指正。

编　　者

1995年10月于华中理工大学

目　录

第一章 引 论

1.1 基础知识

1.1.1 计算机系统

自1946年世界上第一台电子计算机诞生以来，电子计算机的发展已经历了电子管、晶体管、集成电路及大规模超大规模集成电路四个阶段。近50年来，虽然计算机的结构有了很大的变化，其功能有了惊人的提高，但对于任何一个完整的计算机系统来说，无论其结构和功能上的差异如何，其组成均包含硬件和软件两大部分。硬件部分是构成计算机系统的物理设备，是整个计算机系统的物质基础；软件部分是使用和管理计算机所需的各种程序及有关的数据和文档，软件是计算机系统的灵魂。

1. 硬件部分

计算机系统的硬件部分包括主机和外部设备。主机由中央处理单元(通常简称为CPU)和主存储器(通常简称为内存)组成，外部设备包括输入设备、输出设备和外部存储设备。输入设备和输出设备有时简称为I/O设备，外部存储设备通常简称为外存。计算机系统的硬件组成如图1.1所示。

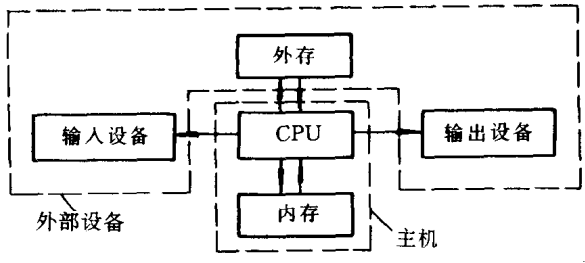


图1.1 计算机系统的硬件组成

1) CPU

CPU是计算机系统的心脏，主要包括运算器和控制器两部分。运算器完成所有的算术运算和逻辑运算；控制器负责从内存取出指令和数据，并执行指令规定的动作以及将执行结果存入内存。

2)内存

内存是存储单元的序列。存储单元是一些仅能表示0和1两种状态的元件，具有这种功能的元件称为一个二进制位(bit)。存储单元以0和1组成的数字串(二进制码)形式存储正在执行的指令和正被处理的数据，这些指令或数据称为存储单元的内容。

存储单元依次按字节(一字节通常为8个二进制位)编址，内存包含的存储单元的总字节数称为内存的容量。

从存储单元取出内容称为读内存，将指令或数据存入内存称为写内存。存取内存或读写内存统称为访问内存。访问内存必须由CPU通过存储单元的地址进行。在高级语言程序中，访

问内存一般是通过程序中的变量进行的，每个变量的名称对应于一定字节数目的存储单元，存储单元的内容就是变量的值。在C程序中，还可以通过存储单元的地址(即内存物理地址)访问内存。

对于一个程序员来说，应了解内存具有的以下重要性质：

(1) 任何时刻计算机一经加电则存储单元中必有内容，但在未通过程序存入内容之前，这些内容仅是随机的，无意义的信息。

(2) 任何时刻计算机一旦切断电源则存储单元中的一切内容立即消失，将不复存在。

(3) 读内存操作永远不改变内存中的内容，即如果仅仅执行读内存的操作，那么，无论读多少次，存储单元中的内容保持不变。

(4) 写内存操作要破坏存储单元中原来的内容。任何时候一旦执行了写内存操作，存储单元中原来的内容立即被新写入的内容所代替，原来的内容不复存在(通常称为被"冲掉")。

3) I/O设备

I/O设备用于人和计算机通信。输入设备用于将程序、各种形态的数据或文档送入计算机内存；输出设备用于将存储在计算机内存中的各种信息送到外部设备上显示以供阅读，或打印出来以供保存。被显示的信息通常为程序的执行结果和程序执行过程中给用户的提示信息；被打印的信息通常为源程序文件、图形和各种文档。

输入设备分为字符设备和图形设备两类。目前最常用的字符输入设备有终端键盘，最常用的图形输入设备有鼠标和图形输入板。输出设备主要包括打印机、终端显示器和绘图仪。打印机和终端显示器可以兼作字符输出设备和图形输出设备，绘图仪只能用于输出图形。其中，终端键盘和终端显示器通常被系统规定为标准输入设备和标准输出设备。

4) 外存

外存既是信息的永久存储设备又起着输入输出设备的作用。由于外存比内存廉价得多，所以外存容量比内存容量大得多，外存容量可以大到以GB(1GB=10^9字节)为度量单位，而内存容量一般以MB(1MB=10^6字节)为度量单位。所以，外存可用于存放大量的程序、数据和文档。

主要的外部存储设备有磁盘驱动器、磁带机和光盘驱动器，内存的信息可以通过这些外部存储设备被写到磁盘盘片、磁带或光盘盘片等存储媒体上。信息一经写入，只要存储媒体未被损坏或未重新写入别的信息，则信息将被永久保存(关掉电源以后信息也不会丢失)。另一方面，保存在外存上的任何程序必须从外存读入内存才能被执行，任何数据必须从外存读入内存才能被处理。换言之，当需要执行外存上的程序或需要处理外存上的数据时，必须首先将它们从外存输入到内存。

2. 软件部分

计算机系统的软件部分包括系统软件和应用软件。

1) 系统软件

系统软件由计算机厂家及软件公司提供，是用于使用和管理计算机的各种程序及相关数据和文档的总称。系统软件主要包括操作系统、语言处理器、服务性程序和数据库管理系统(DBMS)等。

(1)操作系统：操作系统是最基本的核心系统软件，用于管理计算机系统的各种资源并控制各种程序的正常执行。

(2)语言处理器：语言处理器是将程序设计语言转变为计算机能够直接识别的机器语言的

翻译程序。例如汇编程序、BASIC 解释程序、C 编译程序和 PASCAL 编译程序。

(3)服务性程序:服务性程序是为用户提供某种特定功能的实用程序。常用的服务程序有建立和修改文本文件的编辑程序,用于程序动态查错和纠错的调试程序以及用于内存和磁盘管理的工具程序。

(4)DBMS:DBMS(数据库管理系统)是能够集中存储和统一管理特定应用范围内所有相关信息,并使之具有最大的数据独立性、一致性和安全性的数据管理程序。

2) 应用软件

应用软件是由用户为使用计算机处理各自的实际问题而开发的具有专门用途的程序。由于计算机的应用极为广泛,因而应用软件多种多样。主要的应用软件有各种 MIS 系统、CAD 系统和过程控制系统。

1.1.2 算法及其表示

1. 算法的基本概念

通常人们把计算机为解决某一问题所需的方法与步骤称为算法。算法可分为两大类:一类是科学计算领域用于处理数值数据的算法,例如求定积分,解高阶方程,求极限等;另一类是数据处理领域用于处理非数值数据的算法,例如分类排序,情报检索,绘图等。算法具有以下重要特性:

(1)有穷性:一个算法包含有限个步骤,即算法经过有限步执行后必须终止。

(2)确定性:算法的每一步规定的动作不能有两种以上的理解,即算法每一步的动作是唯一的。

(3)有输入:一个算法有一个或多个输入。输入是执行算法时所需的信息,包括被算法处理的对象和执行的控制信息。

(4)有输出:一个算法有一个或多个输出,输出是算法执行的结果。

(5)有效性:算法的每一步所规定的动作都能有效地执行。例如,一个数被零除就不能有效地被执行。

2. 算法的表示

把算法用一种适当的方式描述出来称为算法的表示。表示算法有多种方法,最基本和最常用的方法是用自然语言和用传统的流程图(通常简称为流程图或框图)描述算法。传统流程图的常用符号及其意义如表 1.1 所示。

3. 算法的表示举例

问题:求 $s=\sum_{n=1}^{10} n^n$

用自然语言描述算法如下:

(1)用变量 s 存放各项的累加和,置初值为 0;用变量 i 作项数计数器,置初值为 1。

(2) 如果 i≤10 则计算 s,否则转步骤(3)。

计算 s:(2)-1 用变量 j 作每一项的累乘次数计数器,置初值为 1。

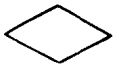
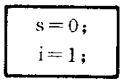
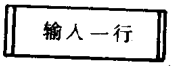
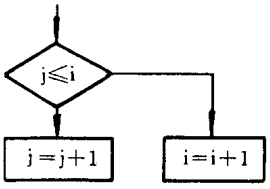
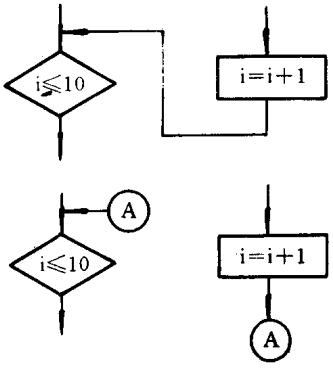
(2)-2 用变量 ai 存放第 i 项的累乘积,置初值为 1。

(2)-3 如果 j≤i 则计算第 i 项,否则转步骤(2)-4。

计算第 i 项:(2)-3-1 ai=ai * i。

(2)-3-2 j=j+1;转步骤(2)-3。

表 1.1　流程图常用符号

符号	符号名称	意　义	例
	起止框	表示算法的开始和结束	
	判断框	表示选择控制：根据框中条件从两种可选的动作中选一执行	
	处理框	表示按顺序执行的处理	
	调用框	表示调用子程序（函数或过程）	
	流程线	表示两个步骤相邻，且执行顺序由箭尾一方到箭头一方。对于自上而下和从左至右的顺序，箭头可省略	
	连接点	连接点必须以相同形式成对出现，用于表示一条流程线被断开后的两个端点	下面两种表示是等价的： 

注：本表所列符号摘自国际信息处理标准 ISO 8631-1986E。

(2)-4 将第 i 项加到累加和中去：

(2)-4-1 s=s+ai。

(2)-4-2 i=i+1；　转步骤(2)。

(3) 输出 s，结束。

上述算法的流程图如图 1.2 所示。

显然，用流程图描述算法比用自然语言描述算法更为简明、直观。

1.1.3　程序设计及程序设计语言

程序是用程序设计语言描述的由计算机解题的算法或计算机解题任务。程序设计是将解题任务转变成程序的过程，主要步骤包括分析问题、确定算法（对复杂算法需画出流程图）、用选定的程序设计语言编写源程序、上机调试和运行程序。程序设计语言是计算机能够理解的、用于人和计算机通信的语言，程序设计语言由低级到高级可分为三类：低级语言，高级语言和

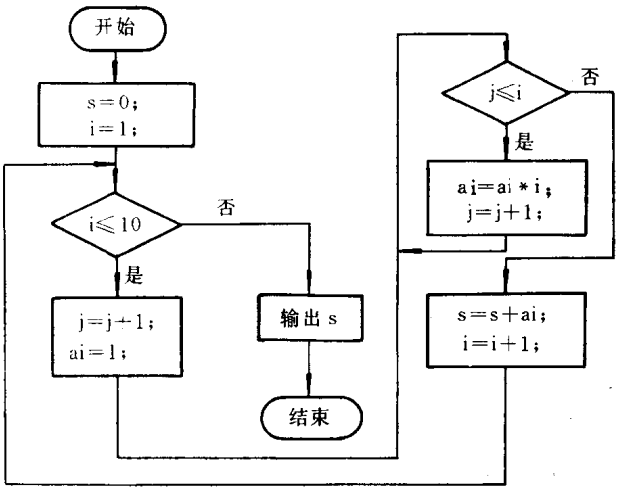


图 1.2 计算 $s=\sum_{n=1}^{10} n^n$ 的算法流程图

专用语言。

1. 低级语言

低级语言又分为机器语言、符号语言和汇编语言。机器语言用二进制代码表示机器指令和数据，机器语言程序能够直接被机器理解和执行，因而程序效率高。但编程繁琐，且不便于记忆和阅读，因而程序维护困难。符号语言用符号代替二进制代码表示机器指令。汇编语言进而用符号代替二进制代码表示指令和数据的存储地址。现在人们用低级语言编程通常指用汇编语言编程。汇编语言程序必须被转换为机器语言程序才能被计算机理解和执行，完成这种转换任务的系统软件称为汇编程序，这种转换过程称为汇编。低级语言是面向机器的，用低级语言写的程序效率高，但没有可移植性，即不能从一个机器系统上移到另一个机器系统上运行。此外，用机器语言编写程序要求程序员必须懂得具体机器系统的硬件结构。

2. 高级语言

高级语言是类似于人类自然语言和数学描述语言的程序设计语言，例如 C 语言和 PASCAL 语言。用高级语言编写的程序称为源程序，通常简称程序，例如 C 源程序或 C 程序，PASCAL 源程序或 PASCAL 程序。高级语言程序也必须被转换成机器语言程序才能够被机器理解和执行，完成这种转换任务的系统软件称为编译程序，例如 C 编译程序和 PASCAL 编译程序，相应的转换过程称为编译。

高级语言是面向解题过程的，语言本身与具体机器系统无关，因而用高级语言编写的应用程序可移植性好。编译程序是一种语言的具体实现，编译程序与具体机器系统有关，常称为某个语言的一个版本。同一语言的不同版本不完全相同，在使用一种具体的高级语言及其编译程序开发软件时，必须参考与编译程序配套的有关资料。本书阐述的内容遵从 ANSI C 标准，对于大多数 C 编译程序具有通用性。为使上机环境尽量简单，书中所有例题均在 Turboc 2.0 上通过。

3. 专用语言

专用语言是为解决特定的应用而设计的计算机程序设计语言。例如，计算机辅助设计(CAD)系统中使用的绘图语言，数据库管理系统(DBMS)的数据查询语言等。专用语言是面向问题的语言，它比高级语言更抽象，描述能力比高级语言更强，用专用语言编程不需要指出“如

何做”，只需说明要“做什么”。

1.2 C语言简况

1.2.1 C语言的发展过程

C语言是目前世界上流行最广泛的高级程序设计语言。C语言的发展过程可粗略地分为三个阶段：1970年至1973年为诞生阶段，1973年至1988年为发展阶段，1988年以后为成熟阶段。

1. C语言的诞生

C语言是为写UNIX操作系统而诞生的。1970年美国AT&T公司贝尔实验室的ken Thompson为实现UNIX操作系统而提出一种仅供自己使用的工作语言，由于该工作语言是基于1967年由英国剑桥大学的Martin Richards提出的BCPL语言设计的，因而被作者命名为B语言，B取自BCPL的第一个字母。B语言被用于在PDP-7计算机上实现了第一个UNIX操作系统。1972年贝尔实验室的Dennis M.Ritchie又在B语言基础上系统地引入了各种数据类型，从而使B语言的数据结构类型化，并将改进后的语言命名为C语言，C取自BCPL的第二个字母。可见，C语言名字的由来反映了C语言诞生所经历的两个过程。1973年K. Tompson和D. M. Ritchie用C语言重写了UNIX操作系统，推出UNIX V5，1975年又推出UNIX V6。此时的C语言是附属于UNIX操作系统的。

2. C语言的发展

为了使UNIX操作系统能够在别的机器上得到推广，1977年C的作者发表了不依赖于具体机器系统的C语言编译文本《可移植C语言编译程序》，从而推动了UNIX操作系统在各种机器上的实现以及UNIX操作系统的不断发展。1978年以后相继推出了UNIX V7，UNIX system V。UNIX操作系统的巨大成功和广泛使用使人们普遍注意到C语言的突出优点，从而又促进了C语言的迅速推广。同时，C语言也伴随着UNIX操作系统的发展而不断发展。1978年Brian W. Kernighan和D. M. Ritchie以UNIX V7中的C编译程序为基础写了影响深远的名著The C Programming Language，这本书上介绍的C语言是以后各种C语言版本的基础，被称为传统C。1978年以后，C语言先后移植到各种大型机、中型机、小型机及微型机上。目前，C语言已成为世界上使用最广泛的高级程序设计语言，且不依赖于UNIX操作系统而独立存在。

3. C语言的成熟

1978年以后，C的不断发展导致了各种C语言版本，不同的C语言版本对传统C都有所扩充和发展。1988年，美国国家标准协会(ANSI)综合了各版本对C的扩充和发展，制定了新的C语言文本标准，称为ANSI C。ANSI C实现了C语言的规范化和统一化。B. W. Kernighan和D. M. Ritchie按ANSI C标准重写了他们的经典著作，于1990年正式发表了The C Programming Language Second Edition。现在人们通常称ANSI C为标准C(1990年国际标准化组织(ISO)公布的C语言标准是以ANSI C为基础制定的)。C语言标准的制定标志着C语言的成熟，1988年以后推出的各种C语言版本对ANSI C是相容的。

1.2.2 C语言的主要特点

C语言之所以成为目前世界上使用最广泛的程序设计语言，并被选作为适应近代软件工程需要而发展起来的面向对象的程序设计语言C++的基语言，是由于C语言的诸多突出优点所决定的。下面从语言本身和应用角度两个方面概括C语言的主要特点。

1. C的语言特点

C的语言特点是表达能力强，流程控制结构化，语言简练且使用灵活。

(1) 表达能力强：C语言有丰富的数据类型和运算符；可以直接访问内存物理地址和硬件寄存器；可以表达直接由硬件实现的针对二进制位的运算。

(2) 流程控制结构化、程序结构模块化：C语言具有各种流程结构的控制语句和多种转移语句，控制语句与适当的转移语句相结合可具有很强的流程控制功能，有助于编制结构良好的程序；此外，使用函数作为程序的基本单位以及变量的存储类属性，在某种程度上实现了数据的隐藏和模块化程序设计。

(3) 语言简练：C语言只有6种基本语句(表达式语句、复合语句、标号语句、选择语句、循环语句和转移语句)；在语言成份的表示方法上尽可能简洁，例如，用%代表MOD表示求余数运算符，/同时兼作整数除运算符和实数除运算符，在if语句中用条件加括号( )代替在条件后面写then关键字，用{和}分别代替begin和end表示复合语句的开始和结束；此外，程序所需的某些操作不是作为C的语法成员而是通过C库函数实现的，例如I/O操作。因而，C程序简洁，C编译程序的体积也小。

(4) 使用灵活：C语言是非强类型的语言，变量及其值在数据类型上不要求具有严格的对应关系。例如，char、int、float和double类型的变量之间可以相互赋值，char类型可以作为int类型数据参加运算，不存在区别于整数的逻辑值(以非零整数表示逻辑真值，整数零表示逻辑假值)。此外，表达式可以作为语句使用；数组的元素和结构的成员可以用指针来表示。

2. C的应用特点

(1) C程序代码质量高：代码质量是指C程序经编译后生成的目标程序在运行速度上和存储空间上开销的大小，运行速度越高、占用的存储空间越少，则代码质量越高。一般高级语言相对于汇编语言而言其代码质量要低得多，但C语言在代码质量上几乎可以与汇编语言媲美。

(2) C程序可移值性好：可移植性是指将一个程序不作改动或稍加改动就能从一个机器系统上移到另一个机器系统上运行。由于C语言独立于具体机器系统以及C语言的标准化，使得用C语言编写的程序包括C编译程序本身可移植性好。

由于C语言的上述语言特点和应用特点，使它成为一个实用的通用程序设计语言，既可用于编写系统软件又可编写应用软件，特别适用于编写各种与硬件环境相关的系统软件，不愧为一种强有力的系统程序设计语言。

C语言的不足之处在于它是一种弱类型语言。这既是C语言的使用灵活之处，也是C语言的不安全因素。对于有经验的C程序员来说，他能够避免不安全因素可能造成的影响而更看重于C使用灵活的优点。