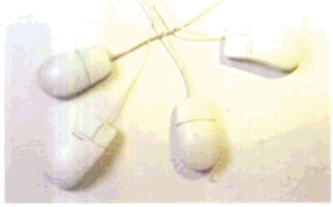
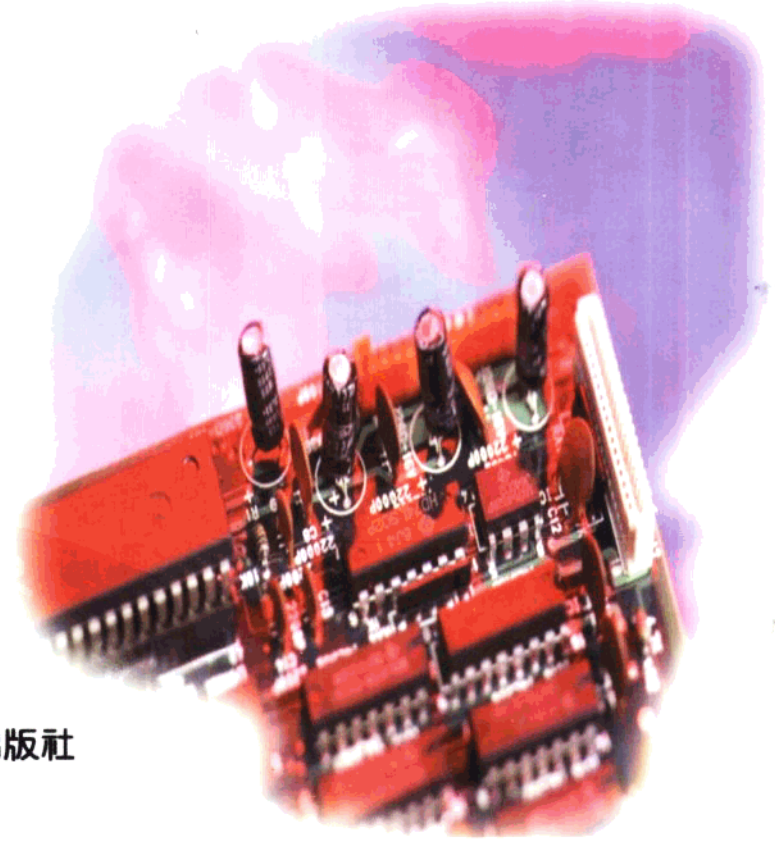


山东省高校统编教材

C/C++语言程序设计

山东省教育厅组编



石油大学出版社

山东省高校统编教材

C/C++语言程序设计

崔培伟　赵合计
宋吉和　贾小珠　于广斌

石油大学出版社

图书在版编目（CIP）数据

C/C++语言程序设计/崔培伟等编.—东营：石油大学出版社，2000.12
ISBN 7-5636-0774-9

Ⅰ.C…　Ⅱ.崔…　Ⅲ.C语言-程序设计　Ⅳ.TP312

中国版本图书馆CIP数据核字（2000）第55426号

C/C++语言程序设计

崔培伟　赵合计
宋吉和　贾小珠　于广斌

出版者：石油大学出版社（山东 东营，邮编 257061）
印刷者：石油大学印刷厂
发行者：石油大学出版社（电话 0546-8392563）
开　本：787×1092　1/16　印张：20　字数：512千字
版　次：2000年12月第1版　2000年12月第1次印刷
印　数：1—5000册
定　价：23.50元

前　言

随着计算机科学技术的迅速发展，所有的计算机语言也在不断地发展和完善。C 语言是近些年来发展、推广速度最为迅速的一门程序设计语言，它的组成简洁、紧凑，使用方便、灵活，运算符和数据结构丰富，处理功能强，目标代码质量高，既具有高级语言的一般特点，又具有汇编语言对硬件和二进制位操作的特殊功能。

结构化程序设计早已被公认为是一种好的程序设计方法，并在各软件开发领域广泛应用，而 C 语言就是最适合于结构化程序设计的一门语言，它的模块化结构强，可移植性好，很适合编写大型软件。面向对象程序设计方法是近几年来发展很迅速、比较新颖的程序设计方法，C++语言是面向对象程序设计方法应用较广、功能较强的语言，它在 C 语言的基础上进行了扩充，加进了面向对象的思想。有了 C 语言的基础，可以较快地理解并掌握 C++语言，掌握面向对象的设计思想和方法。

本教材前八章旨在使读者掌握结构化程序设计的思想方法，并学会用 C 语言作为工具来具体实现和解决实际问题。因此，一开始便从算法设计入手，并始终贯穿结构化思想，以使初学计算机语言的读者在学习与实践中逐步养成良好的习惯，建立良好的程序设计风格。第九、十一章主要介绍如何用 C++语言实现面向对象的程序设计，逐步掌握面向对象程序设计的方法，并用 C++语言进行面向对象的程序设计。

全书共分十一章。第一、四、五章由石油大学崔培伟执笔，第二、三章及附录由石油大学于广斌执笔，第六、七章由山东大学赵合计执笔，第八、十章由山东工程学院宋吉和执笔，第九、十一章由青岛大学贾小珠执笔，全书由崔培伟策划并修改定稿。在教材编写过程中，李宗民老师给予了很多宝贵建议，在此表示衷心的感谢。

作者在编写过程中主观愿望是力求使概念叙述准确，方法条理清楚，并力争用最少的篇幅把最有用、最新颖的内容介绍给读者，但由于水平所限，书中难免存在缺点错误，恳请专家、同行和广大读者批评指正。谢谢！

作　者

2000 年 10 月

目 录

第一章 概 述

§1.1 程序设计和程序设计语言

1.1.1 程序、程序设计、程序设计语言的概念

“程序”这个词大家都不陌生，做任何事情都有个“程序”。比如，开会有“程序”（叫会议议程）：大会进行第一项——奏国歌；大会进行第二项——XXX 讲话；……。厨师炒菜也有程序：①放油；②放葱花和盐；③放菜。这个程序称为菜谱。那么，对计算机而言，程序是什么呢？人们要让计算机解决一个问题时，需把解决这个问题的步骤通过一条条指令的形式告诉计算机，一般我们把人们事先准备好的、用来指挥计算机工作的描述工作步骤的指令序列称为**程序**。程序员设计编写程序的过程我们称为**程序设计**。用来编写程序的语言称为**程序设计语言**。

1.1.2 程序设计语言的发展

1. 机器语言

最初的阶段，人们直接使用计算机能识别的指令系统（称为机器语言）来编写程序，由于机器语言是二进制代码，人们编写或阅读程序都十分困难，又容易出错，且不同机器的指令系统也不同，很难进行交流（在一种机器上调试通过的程序不能到另外一种机器上运行）。机器语言程序虽然其执行效率很高，但花费在程序设计和调试程序上的时间太多，整个解决问题的效率就降低了。

2. 汇编语言

为了解决二进制代码编程带来的困难，人们采用了助忆码和符号地址来代替机器语言中的二进制指令代码和指令地址，然后通过一个人们预先设计好的叫做“汇编程序”的翻译程序一对一地翻译成机器语言程序，再让计算机执行。这种采用助忆码和符号地址的语言称为**汇编语言**。用汇编语言编写的程序执行效率与机器语言程序一样高，且其阅读性提高了，但由于汇编指令与机器指令之间是一对一的，同样不利于交流，所以，汇编语言也是面向机器的语言。

3. 高级语言

随着计算机各种技术的发展，程序设计语言也在不断发展，为了脱离机器，为了非计算机专业人员的使用，人们开始用一种比较接近于自然语言（主要指英语）和数学语言的语言来编写程序，这样的语言称为**高级语言**。高级语言脱离了计算机的具体指令系统，对于非计算机专业人员来说掌握起来比较容易，克服了面向机器语言的缺点，使得程序易读、易维护、易交流。高级语言发展很快，已达数百种之多，常用的高级语言有：

（1）FORTRAN 语言。诞生于 20 世纪 50 年代中期，是第一个算法语言，适应于科学和工程计算。

（2）BASIC 语言。诞生于 20 世纪 60 年代中后期，语言简单易学，是一种会话型语言，

适合初学者学习。

（3）PASCAL 语言。诞生于 20 世纪 70 年代初，是一门结构化程序设计语言，适合于教学、科学计算、数据处理和系统软件开发。随着 C 语言的出现，逐步被取代。

（4）C 语言。诞生于 20 世纪 70 年代初，80 年代开始风靡全世界，适应于系统软件、数值计算、数据处理。

（5）JAVA 语言。诞生于 20 世纪 90 年代，是一种新型的跨平台分布式程序设计语言，具有简单、安全、稳定、可移植性强等特性，将成为未来网络环境上的“世界语”。

一般地，把用高级语言或汇编语言编写的程序称为**源程序**。

1.1.3 源程序在计算机上的执行过程

人们用高级语言编写的源程序计算机并不懂，像汇编语言一样，必须经过一个“翻译”将高级语言编写的源程序翻译成计算机能识别的二进制代码程序，才能让计算机执行并得到结果。这个“翻译”一般称为**语言处理程序**。

语言处理程序将源程序翻译成二进制代码的方式通常有两种，一种称为“编译方式”，一种称为“解释方式”。

1．编译方式

编译方式将源程序全部翻译成一个功能等价的机器语言程序（一般称为“目标程序”），然后经过一个“连接程序”将用户的目标程序（一个或多个）与系统配置好的一些通用程序连接装配在一起，形成一个“可执行程序”，最后让计算机执行并得到结果。

编译方式经编译连接得到的可执行程序可以重复执行无数次，其效率很高。系统程序、用户已调试好的程序，大都采用编译方式处理，最终得到可执行程序供使用。

2．解释方式

解释方式是通过“解释程序”逐句翻译、执行的，不产生目标代码程序，每次执行都要重新翻译，所以对那些重复执行的程序效率较低。但解释方式对于调试程序很方便。

目前，大多数常用的高级语言都采用编译方式，且输入或修改源程序、编译、连接、执行四个过程均在一个集成环境下，省去了用户不少的麻烦。

§1.2 算法及其表示

1.2.1 算法

前面我们已讨论过，做任何事情都有一定的方法步骤，步骤错了就要出问题。如果炒菜时先放菜后放油，炒出的菜肯定不好吃。同样，让计算机进行计算的步骤搞错了，结果就会出问题。设想控制发射卫星的计算机程序如果出了问题，后果会怎么样？所以工作步骤的描述是程序设计中关键的一环。通常我们把这种具体工作步骤的描述称为**算法**。实际上程序就是算法在计算机上的实现。

计算机能否正确有效地工作，很大程度上取决于人们在程序设计时是否能设计一个好的算法。那么，什么样的算法才是一个好的算法呢？一个好的算法除了满足有穷性（操作步骤有限）、正确性要求之外，还应当具备结构性好、效率高两个特点。所谓结构性好就是算法要结构清晰、步骤简单、容易阅读理解和容易在计算机上实现。而效率高一般指解决一个

问题所花费的计算机运行时间和存储空间尽量少。

例 1.2.1 设计一个计算 $S=\sum_{n=1}^{100} n$ 的算法。

算法一：设存储单元 S 为累加单元。

Step1 将 1 送到 S 单元中；

Step2 把 2 加到 S 中（即取 S 中的内容加 2 后再送回 S 单元中）；

Step3 把 3 加到 S 中；

Step4 把 4 加到 S 中；

Step5 把 5 加到 S 中；

… …

Step99 把 99 加到 S 中；

Step100 把 100 加到 S 中；

Step101 把 S 中的结果输出。

算法二：求和操作要重复若干次（此处为 100 次），我们可以设一个计数器 n，每重复一次 n 增 1，直到 n 大于 100 为止，求和操作可以改为“n+S 送 S”。修改后的算法如下：

Step1 将 0 送到 S 中；

Step2 将 1 送到 n 中；

Step3 把 n 的值加到 S 中；

Step4 n 增 1；

Step5 若 n≤100 则转回 Step3，否则执行 Step6；

Step6 输出 S 的值。

显然，算法二比算法一简单、实用、修改方便。若求和的起始值和终止值改变（如改为 10 和 200）了，算法二只需修改 Step2（1 改为 10）和 Step5（100 改为 200）即可，而算法一却要全部修改。

例 1.2.2 有两个单元 a 和 b，要求将它们的值互换。

按存储器的性质，如果将单元 a 的值直接送到单元 b 中，那么就会冲掉 b 原来的内容，因此，需要一个临时单元 c 来缓冲一下（见图 1-1）。

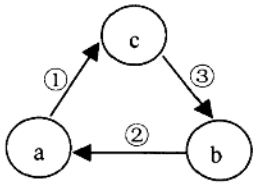


图 1-1 交换两单元内容的图示

交换两单元内容的算法如下：

Step1 将单元 a 的值送给单元 c；

Step2 将单元 b 的值送给单元 a；

Step3 将单元 c 的值送给单元 b。

例 1.2.3 找出十个数中的最大数。

本题思路：同擂台赛一样，挑战者先站在擂台上，第二个人与之比较，败者下，胜者留，第三个人再上台与台上的人（胜者）比较，同样败者下胜者留，……，第十个人与台上的人比较，败者下胜者留，最后台上的人为胜者，这样共进行九轮。我们用单元 max 记最大的数（开始为第一个数），用 n 作计数器，其他数用 x 记，如果 x>max 则 x 送到 max 中。算法如下：

Step1 输入第一个数送单元 max。

Step2 计数器置 1（1 送 n）。

Step3 当 n≤9 时重复下述操作：

（1）输入下一个数 x；

（2）若 x>max 则 x 送 max；

（3）n 增 1。

Step4 输出 max 的值。

例 1.2.4 用加减法求两个正整数 x 和 y 的整数商和余数。

x 与 y 的整数商其含义是求 x 中有几个 y，我们可以重复操作“x-y 送 x”直到 x<y 为止，重复几次，商就是几，而 x 中剩下的值就是余数。这样就需要一个计数器 count 来记重复次数。本题的算法如下：

Step1 计数器 count 置 0；

Step2 输入 x 和 y；

Step3 x-y 送 x；

Step4 计数器加 1；

Step5 若 x≥y 则返回 Step3，否则执行 Step6；

Step6 输出 count 和 x 的值。

例 1.2.5 求 n!。

设单元 p 存放阶乘的值，根据阶乘的含义 p=n×(n-1)×(n-2)×…×3×2×1，本题与例 1.2.1 类似，只是加变为乘而已。算法如下：

Step1 将 1 送到 p 中。

Step2 将 1 送到 k 中。

Step3 输入 n。

Step4 当 k≤n 时重复下面的操作：

（1）p×k 送 p；

（2）k 增 1。

Step5 输出 p 的值。

例 1.2.6 判断一个正整数 n 是否为素数。

所谓素数就是除了 1 和本身之外，不能被任何数整除的数。判断一个数是否素数最直观的办法就是用 2、3、4、…、(n-1）去除 n，只要其中有一个能整除 n，则 n 不是素数。所以，否定一个数不是素数很容易，而肯定一个数是素数需排除所有可能的因子。设 flag 是一个标志，其值为 0 时表示 n 不是素数，非 0 时表示 n 是素数。由此思路得到算法如下：

Step1 输入 n 值。

Step2 flag 置 1（先假设 n 是素数）。

Step3 令 m 为 2。

Step4 当 m≤n-1 时重复下述操作：

（1）若 m 整除 n 则 flag 置 0；

（2）m 增 1。

Step5 若 flag 非 0 则输出“n is prime number”；否则，输出“n is not prime number”。

其实 n 的最大因子为 n/2，m＞n/2 后不可能再整除 n，所以 Step4 中的判断条件可以改为 m≤n/2。用数学方法还可以证明判断条件改为 m≤$\sqrt{n}$ 即可。

例 1.2.7 用计算机验证著名的 Nicomachs 定理：任何一个正整数的立方都可以写成一串奇数之和。

分析：$1^3=1$

$2^3=3+5=8$

$3^3=7+9+11=27$

$4^3=13+15+17+19=64$

$5^3=21+23+25+27+29=125$

…

以上各式有如下规律：

（1）n^3 是 n 个奇数之和；

（2）这 n 个奇数是相邻的，只要确定了一个，其他都确定了。

（3）n 个奇数中最大的奇数是第 m 个（$m=1+2+3+\cdots+n=\frac{n(n+1)}{2}$）奇数，其值为 max=2m-1；其他项依次少 2。

用 k 记其中的奇数项，开始则为 max，以后每次减 2；用 count 作计数器，sum 作累加器，算法如下：

Step1 输入要验证的正整数 n（n≥1）；

Step2 计数器置 1；

Step3 $\frac{n(n+1)}{2}$ 送 m；

Step4 2m-1 送 max；

Step5 max 送 sum；

Step6 max 送 k；

Step7 当 count＜n 时，重复下述动作：

k-2 送 k； k+sum 送 sum； count 增 1；

Step8 如果 sum 与 m 相同则输出“Right”，否则输出“Error”；

Step9 结束。

例 1.2.8 求一元二次方程 $ax^2+bx+c=0$ 的实数根。

设 $d=b^2-4ac$，$d2=\sqrt{b^2-4ac}$，p=2a，算法如下：

Step1 输入 a，b，c。

Step2 求 d。

Step3 求 p。

Step4 若 d＜0 则无实数根，结束。

否则若 d＞0，则：

（-b+d2）/p 送 root1；

（-b-d2）/p 送 root2。

否则：

-b/p 送 root1；

root1 送 root2。

Step5 输出 root1 和 root2。

例 1.2.9 有三个不为零的数字，它们可能组合成的所有三位数的和等于 2886。如果把这三个数从大到小和从小到大排列成两个三位数，其差等于 495。试求出这三个数。

分析：设这三个数字从大到小分别为 a、b、c，用它们可能组成的三位数是：

100a+10b+c

100c+10b+a

100c+10a+b

100a+10c+b

100b+10a+c

100b+10c+a

它们的和为 222（a+b+c）=2886。

又因为它们从大到小和从小到大排列的差为 495，即

（100a+10b+c）-（100c+10b+a）=495

所以 a=c+5。因为 a>b>c>0，所以 a 最大值为 9，c 最大值为 4（取 1～4），b 取 c+1～c+4。对每一组 a、b、c，若 222（a+b+c）的值为 2886，则 a、b、c 为所求。根据分析得到算法如下：

Step1 令 c 为 1。

Step2 当 c≤4 时重复下述操作：

（1）令 a 为 c+5；

（2）令 b 为 c+1；

（3）当 b≤c+4 时重复下述操作：

① 若 222（a+b+c）等于 2886 则输出 a、b、c；

② b 增 1；

（4）c 增 1。

Step3 结束。

1.2.2 算法的表示

虽然用程序设计语言表示算法是程序设计的最终目的，但由于程序设计语言是一种形式化语言，要求严格，在初学阶段或设计比较复杂的问题时很难直接使用，所以，要寻找几种直接表示算法的方法。下面我们介绍几种算法的表示方法。

1．用自然语言描述

前面我们给出的算法全部采用自然语言描述，这种方法通俗易懂，但有它的缺点：

（1）不太严格且繁琐冗长；

（2）用自然语言描述顺序执行的步骤比较好懂，但如果算法复杂且包含很多判断转移（如例 1.2.9）时，用自然语言描述就不是那么直观清晰了。

2. 算法框图

作为算法的表达工具，在算法设计阶段人们一般采用图示的方法。图示法简单、直观，特别有利于初学者。算法的图示法又称为算法框图。有两种形式的算法框图：

（1）传统的程序流程图：

传统的程序流程图通常采用一些几何图形来代表各种类型的操作，并在图形内标明文字或符号，表示操作的内容，用箭头来表示操作的顺序。图 1-2 给出了流程图中使用的图形符号及代表的含义。图 1-3 ～ 图 1-6 分别为前面所给例题的流程图（注：流程图中我们用 0⇒S 表示“将 0 送 S”或“S 置 0”）。

图 1-2 程序流程图常用符号

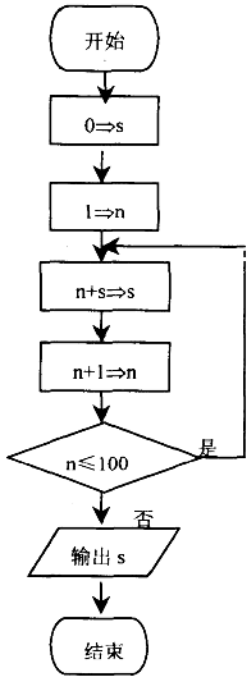


图 1-3 求 $S=\sum_{n=1}^{100} n$ 的流程图

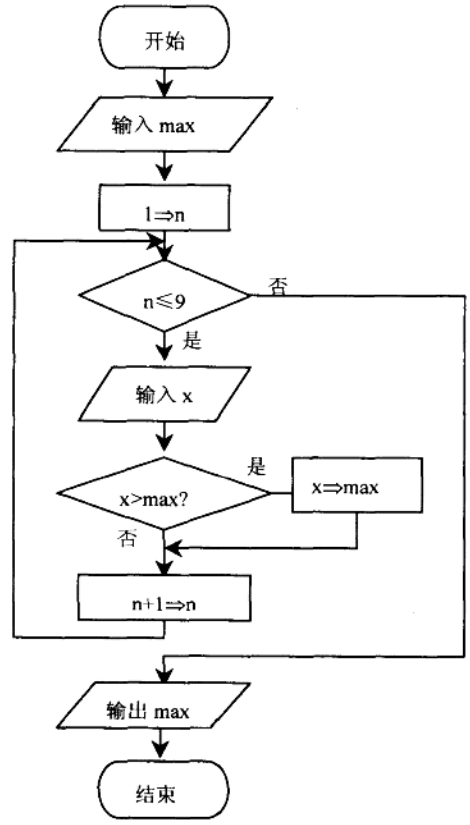


图 1-4 找出 10 个数中的最大数