高 职 高 专 计 算 机 系 列 教 材

C语言程序设计

冉崇善 白涛 李思辉 刘炜 贾小云 吴华 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机系列教材

C 语言程序设计

冉崇善　白　涛　李思辉
刘　炜　贾小云　吴　华　编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本教材是高职高专学生学习C语言程序设计的理想教材。全书共分为9章，主要内容包括：程序设计与算法，数据类型、运算符与表达式，程序控制语句，数组，函数，指针，结构体与共用体，输入输出和文件系统，程序设计例解等。全书以ANSI C语言标准为基础，以C语言程序设计为主线，介绍了程序设计的基本概念、C语言的语法规则和实用的C程序设计技术。本书结合应用实例，强调“好的”C程序编写方式，力图展示给初学者一个良好的程序设计入门向导。

本教材在结构上突出了以程序设计为中心、以全国计算机等级考试（二级C语言）为主线，深入浅出地介绍了程序设计在实际中的应用。在内容上注重知识的完整性，以适合初学者的需求。

本教材既可作为高职高专学校非计算机专业学生的C语言学习教材，又可作为全国计算机等级考试（二级C语言）的参考教材，还可作为科技人员自学C语言的参考书。

图书在版编目（CIP）数据

C语言程序设计/冉崇善等编著. —北京：中国铁道出版社，2007.1

（高职高专计算机系列教材）

ISBN 978-7-113-07775-4

Ⅰ.C… Ⅱ.冉… Ⅲ.C语言－程序设计－高等学校：技术学校－教材 Ⅳ.TP312

中国版本图书馆CIP数据核字（2006）第160770号

书　　名：C语言程序设计

作　　者：冉崇善　白　涛　李思辉　等

出版发行：中国铁道出版社（100054，北京市宣武区右安门西街8号）

策划编辑：严晓舟　秦绪好

责任编辑：苏　茜　翟玉峰　郑　楠

封面设计：薛　为

封面制作：白　雪

责任校对：李　旸

印　　刷：河北省遵化市胶印厂

开　　本：787×1092　1/16　**印张：**18.25　**字数：**421千

版　　本：2007年2月第1版　　2007年2月第1次印刷

印　　数：1～5 000册

书　　号：ISBN 978-7-113-07775-4/TP·2140

定　　价：24.00元

高职高专计算机系列教材

编　委　会

前　言

C语言作为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展现出了强大的生命力。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，又适合于开发人员编写应用软件，是计算机应用人员应掌握的基本功。所以长久以来，C语言广泛流行，经久不衰。

针对许多人想学C语言而又感到C语言难学的实际情况，作者在多年C语言教学的基础上，编写并出版了这本《C语言程序设计》教材。该书采用了新的体系结构，分散难点，减小台阶，用人们易于理解的方式叙述复杂的概念，可使高职高专学校的学生、参加全国计算机等级考试的考生感到C语言不再像人们想象的那样难学了。

全书共分为9章，主要内容包括：程序设计与算法，数据类型、运算符与表达式，程序控制语句，数组，函数，指针，结构体与共用体，输入输出和文件系统，程序设计例解等。全书以ANSI C语言标准为基础、以C语言程序设计为主线，介绍了程序设计的基本概念、C语言的语法规则和实用的C程序设计技术。本书结合应用实例，强调“好的”C程序编写方式，力图展示给初学者一个良好的程序设计入门向导。

本教材在结构上突出了以程序设计为中心、以全国计算机等级考试二级C语言大纲为主线、以语言知识为工具的思想，对C语言的语法规则进行了整理和提炼，深入浅出地介绍了它们在程序设计中的应用；在内容上注重知识的完整性，以适合初学者的需求。例如，在第1章安排了程序设计与算法，用于介绍程序设计的基本概念、算法、数据结构、流程图、C语言环境和上机调试程序等内容。在第9章安排了程序设计例解，通过较为详细的算法分析和编程思想的介绍，会给读者进一步开发大型软件奠定较为扎实的基础。在写法上追求循序渐进、通俗易懂，旨在引导初学者入门。

本教材以实际应用为目的，侧重于C语言的基本知识，着重讲解概念，深入浅出地讲解用计算机解决问题的方法；内容编排体系合理、逻辑清楚、例题丰富、通俗易懂；同时又根据C语言新标准的规定，非常符合当前的需要。

本书中介绍的C语言，覆盖了2004版《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”。

C语言程序设计是一门实践性很强的课程，不是只靠听课或看书就能掌握的。读者应当十分重视自己动手编写程序和上机运行程序能力的培养，多多进行上机实践。

全书各章配有习题，并有配套的习题解析及上机指导参考书。全部习题均以全国计算机等级考试二级C语言笔试和上机考试的题型或原题作为原型，所有例题和习题均已在Turbo C环境下调试、运行。

本教材由陕西科技大学计算机与信息工程学院冉崇善副教授拟定了编写大纲和框架结构，并编写了3、7、9章和附录，贾小云编写了第2章、吴华编写了第8章、白涛编写了第1、4章、刘炜编写了第5章、李思辉编写了第6章，最后由冉崇善统一定稿。

为了帮助读者学习并理解本书，我们还编写了与之配套的《C 语言程序设计习题解析与上机实验指导》一书，提供《C 语言程序设计》教材中各章习题解答、上机实习指导和等级考试习题解析。本书是高职高专学生学习 C 语言程序设计的理想教材，凡具有计算机初级知识的读者都能读懂本书。本书也可作为参加全国计算机等级考试二级 C 语言考试的培训教材，还可供 C 语言爱好者自学参考。

编　者

2006 年 11 月

目　录

第 1 章　程序设计与算法

计算机是一种以二进制数据形式在内部存储信息、以程序存储为基础、由程序自动控制的电子设备。

计算机程序是由人事先规定的计算机完成某项工作的操作步骤（算法）。每一步骤的具体内容由计算机能够理解的指令来描述，这些指令告诉计算机“做什么”和“怎样做”。

程序设计则是用计算机语言（程序设计语言）实现算法的过程。所以说程序设计离不开算法，算法用来指导程序设计，是程序的灵魂。

1.1　计算机语言和程序

1.1.1　计算机语言的概念

计算机语言是计算机能够理解和识别的语言，是人与计算机进行信息交流的工具。它通过一定的方式向计算机传送操作指令，从而使计算机能够按照人们的意愿进行各种操作处理。任何一种计算机语言都有一定的使用规则，通常称之为语法规则。

要学习计算机语言，必须注意学习它的语法规则，就像学汉语要学汉语语法一样。而学习计算机语言的目的是为了设计计算机程序。

计算机语言的种类有很多，大体上经过了由低级语言到高级语言的发展过程，目前广泛使用的有 C、Visual Basic、Visual FoxPro、Delphi、C++、C#、Java 等。

1.1.2　C 语言的特点

1. C 语言是中级语言

C 语言通常称为中级计算机语言。中级语言并没有贬义，不意味着它功能差、难以使用，或者比 BASIC、Pascal 那样的高级语言原始，也不意味着它与汇编语言相似，会给用户带来类似的麻烦。C 语言之所以被称为中级语言，是因为它把高级语言的成分同汇编语言的功能结合起来了。从表 1-1 中可看出 C 语言在计算机语言中所处的地位。

表 1-1　C 语言在计算机语言中的地位

级　别	语　言
高级	C++、Java、Delphi、C#、Visual Basic、Visual FoxPro
中级	C、FORTH、Macro-assembler
低级	Assembler

作为中级语言，C 语言允许对位、字节和地址这些计算机功能中的基本成分进行操作。C 语言程序非常容易移植。可移植性表示为某种计算机写的软件可以用到另一种计算机上去。举例来说，如果为苹果机写的一个程序能够方便地改为可以在 IBM PC 上运行的程序，则称为是可移植的。

所有的高级语言都支持数据类型。一个数据类型定义了一个变量的取值范围和可在其

上操作的一组运算。常见的数据类型是整型、字符型和实数型。虽然 C 语言有 5 种基本数据类型，但与 Pascal 或 Ada 相比，它却不是强类型语言。C 程序允许几乎所有的类型转换。例如，字符型和整型数据能够自由地混合在大多数表达式中进行运算。这在强类型高级语言中是不允许的。

C 语言的另一个重要特点是它仅有 32 个关键字，这些关键字就是构成 C 语言的命令。

2. C 语言是结构化语言

虽然从严格的学术观点上看，C 语言是块结构（Block-Structured）语言，但是它还是常被称为结构化语言。这是因为它在结构上类似于 ALGOL、Pascal 和 Modula-2（从技术上讲，块结构语言允许在过程和函数中定义过程或函数。用这种方法，全局和局部的概念可以通过“作用域”规则加以扩展，“作用域”管理变量和过程的“可见性”。因为 C 语言不允许在函数中定义函数，所以不能称之为通常意义上的块结构语言）。

结构化语言的显著特征是代码和数据的分离。这种语言能够把执行某项特殊任务的指令和数据从程序的其余部分分离出去、隐藏起来。获得隔离的一个方法是调用使用局部（临时）变量的子程序。通过使用局部变量，我们能够写出对程序其他部分没有副作用的子程序。这使得编写共享代码段的程序变得十分简单。如果开发了一些分离很好的函数，在引用时我们仅需要知道函数做什么，不必知道它如何做。切记：过度使用全局变量（可以被全部程序访问的变量）会由于意外的副作用而在程序中引入错误。

结构化语言比非结构化语言更易于程序设计，用结构化语言编写的程序的清晰性使得它们更易于维护。这已是人们普遍接受的观点了。C 语言的主要结构成分是函数 C 的独立子程序。

在 C 语言中，函数是一种构件（程序块），是完成程序功能的基本构件。函数允许一个程序的诸任务被分别定义和编码，使程序模块化。可以确信，一个好的函数不仅能正确工作且不会对程序的其他部分产生副作用。

1.1.3 程序的概念

从生活的角度来说，程序通常是指完成某些事务的一种既定方式和过程。

从计算机的角度来说，程序就是按照计算机语言规则组织起来的一组指令，是为完成某项功能所需要执行的命令序列。这些命令序列按照一定的结构合理地、有机地组合在一起，并以文件的形式存储在磁盘上，故称为命令文件。

1.2 算法

程序的实质是数据结构与算法的结合。对于面向对象的程序设计语言，强调的是数据结构，而对于面向过程的程序设计语言（如 C、Pascal、FORTRAN 等语言），主要关注的是算法。掌握算法，也是为面向对象的程序设计打下一个扎实的基础。

1.2.1 算法的概念

人们使用计算机，就是要利用计算机处理各种不同的问题，而要做到这一点，人们就必须事先对各类问题进行分析，制定出解决问题的具体方法和步骤，再具体编制好一组让

计算机执行的指令（即程序），让计算机按人们编制的指定步骤运行这些指令。这些具体的方法和步骤，其实就是解决一个问题的算法。根据算法，依据某种规则编写计算机执行的命令序列，就是编制程序，而书写时所应遵守的规则，即为某种语言的语法。

由此可见，程序设计的关键之一，是解题的方法与步骤，即算法。学习高级语言的重点，就是掌握分析问题、解决问题的方法，就是培养读者分析、分解，最终归纳整理出算法的能力。与之相对应，具体语言，如 C 语言的语法是工具，是算法的一个具体实现。所以在高级语言的学习中，一方面应熟练掌握该语言的语法，因为它是算法实现的基础，另一方面必须认识到算法的重要性，加强思维训练，以写出高质量的程序。

综上所述，人们将为解决某一个问题而采取的有效、科学的方法与步骤称为“算法”。不要认为只有“计算”的问题才有算法。

1.2.2　算法的特性

对同一个问题，可以有不同的解题方法和步骤。方法有优劣之分，有的方法只需进行很少的步骤，而有些方法则需要较多的步骤。一般来说，希望采用简单和运算步骤少的方法。因此，为了有效地进行解题，不仅需要保证算法正确，还要考虑算法的质量，选择合适的算法。

算法应当具备以下几个方面的特性：

（1）有穷性。一个算法必须保证执行有限步骤之后结束。

（2）确定性。算法的每一个步骤必须具有确切的定义。

（3）输入性。应对算法给出初始量。

（4）输出性。算法具有一个或多个输出。

（5）可行性。算法的每一步骤都必须是计算机能进行的有效操作。

1.2.3　算法的设计

下面通过例子来介绍如何设计一个算法。

【例 1-1】输入三个数，然后输出其中最大的数。

首先，需先有个地方存储这三个数。我们定义三个变量 A、B、C，将三个数依次输入到 A、B、C 中，另外，再准备一个 Max 存储最大数。由于计算机一次只能比较两个数，我们首先把 A 与 B 比，大的数放入 Max 中，再把 Max 与 C 比，又把大的数放入 Max 中。最后，把 Max 输出，此时 Max 中存储的就是 A、B、C 三数中最大的一个数。

算法表示如下：

（1）输入 A、B、C。

（2）A 与 B 中大的一个放入 Max 中。

（3）把 C 与 Max 中大的一个放入 Max 中。

（4）输出 Max，Max 即为最大数。

其中的第 2、3 两步仍不明确，无法直接转化为程序语句，可以继续细化。

（2）把 A 与 B 中大的一个放入 Max 中，若 A>B，则 Max←A；否则 Max←B。

（3）把 C 与 Max 中大的一个放入 Max 中，若 C>Max，则 Max←C。

最后算法可以写成：

（1）输入 A、B、C。

（2）若 A>B，则 Max←A；否则 Max←B。

（3）若 C>Max，则 Max←C。

（4）输出 Max，Max 即为最大数。

这样的算法已经可以很方便地转化为相应的程序语句了。

【例 1-2】猴子吃桃问题。有一堆桃子不知数目，猴子第 1 天吃掉一半，觉得不过瘾，又多吃了一只，第 2 天照此办理，吃掉剩下桃子的一半另加一个，天天如此，到第 10 天早上，猴子发现只剩一只桃子了，问这堆桃子原来有多少个？

此题粗看起来有些无从着手的感觉，那么怎样开始呢？假设第 1 天开始时有 a1 只桃子，第 2 天有 a2 只，…，第 9 天有 a9 只，第 10 天是 a10 只，在 a1，a2，…，a10 中，只有 a10=1 是知道的，现要求 a1，而我们可以看出，a1，a2，…，a10 之间存在一个简单的关系：

```
a9=2*(a10+1)
a8=2*(a9+1)
    …
a1=2*(a2+1)
```

也就是 ai=2*(a (i+1) +1)，i=9，8，7，6，…，1。

这就是此题的数学模型。

再考察上面从 a9～a1 的计算过程，这其实是一个递推过程，这种递推的方法在计算机解题中经常用到。另一方面，这 9 步运算从形式上完全一样，不同的只是 i 的值而已。由此，我们引入循环的处理方法，并统一用 a0 表示前一天的桃子数，a1 表示后一天的桃子数，将算法改写如下：

（1）a1=1；｛第 10 天的桃子数，a1 的初值｝

i=9。｛计数器初值为 9｝

（2）a0=2*(a1+1)。｛计算当天的桃子数｝

（3）a1=a0。｛将当天的桃子数作为下一次计算的初值｝

（4）i=i-1。

（5）若 i>=1，转（2）。

（6）输出 a0 的值。

其中步骤 2～5 为循环。

这就是一个从具体到抽象的过程，具体方法如下。

（1）如果由人来做，应该采取哪些步骤。

（2）对这些步骤进行归纳整理，抽象出数学模型。

（3）对其中的重复步骤，通过使用相同变量等方式求得形式的统一，然后简练地用循环解决。

算法的描述方法有自然语言描述、伪代码、流程图、N-S 图、PAD 图等。

1.2.4 算法的描述方法

算法是实现某一个问题求解的框架流程，而程序设计则是根据这一求解的框架流程进行语言细化并实现这一问题求解的具体过程。

为了使算法表达得更清晰，编写更容易，在程序设计时通常使用专门的算法表达工具

对算法进行描述。如流程图，N-S 图，PAD 图、伪码等。复杂的问题，可以先用算法表达工具对算法进行描述，再进行编程，算法的最终实现应该是计算机程序。算法的评价标准涉及很多方面，但正确性和清晰易懂性永远是一个好算法的基本条件。

描述算法的常用工具有以下几种。

1．自然语言

使用人们日常进行交流的语言。

例如，从 a、b 中找出一个大的数给 max：

（1）从键盘输入两个数给 a 和 b；

（2）如果 a 比 b 大，则把 a 的值传给 max，否则把 b 的值传给 max；

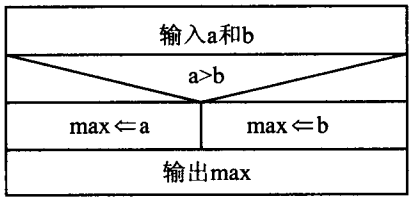
（3）输出 max 的值。

2．专用工具

借助于有关图形工具或代码符号来描述。

常用的工具有流程图、N-S 图等。

如用 N-S 图来描述从 a 和 b 中找大数的问题，如图 1-1 所示。




```
scanf("%d,%d",&a,&b);
if(a>b) max=a;
else max=b;
printf("%d,%d",a,b);
```

图 1-1　N-S 图

3．部分常用的算法

算法最终要用程序设计语言来描述，计算机才能保存、翻译和执行。如用 C 语言来描述从 a、b 中找大数的问题。

常用的算法有迭代法、枚举法、递归法、递推法等。

【例 1-3】求 1×2×3×4×5 的值。

可以用最原始的方法进行：

（1）先求 1×2，得到结果 2。

（2）将步骤 1 得到的乘积 2 再乘以 3，得到结果 6。

（3）将 6 再乘以 4，得 24。

（4）将 24 再乘以 5，得 120，这就是最后的结果。

这样的算法虽然是正确的，但太烦琐。如果要求 1×2×…×1 000，则要写 999 个步骤，显然是不可取的。而且每次都直接使用上一步骤的数值结果（如 2，6，24 等），也不方便。应当找到一种通用的表示方法。

可以设两个变量，一个变量代表被乘数，一个变量代表乘数。不另设变量存放乘积的结果，而直接将每一步骤的乘积放在被乘数变量中。如设 p 为被乘数，i 为乘数，用循环算法来求结果，可将算法改写如下。

S1：使 p=1。

S2：使 i=2。