21世纪
高职高专规划教材系列

C语言程序设计教程

李 敏 主编
刘 婷 陈 双 等参编

增值回报
电子教案

机械工业出版社
CHINA MACHINE PRESS

21 世纪高职高专规划教材系列

C 语言程序设计教程

李 敏 主编

刘 婷 陈 双 等参编

机械工业出版社

本书从实用的角度出发，深入浅出地介绍了 C 语言程序设计的基本概念和方法。编写中参考了《全国计算机等级考试二级考试大纲》中有关 C 语言程序设计的要求，并提供了大量实例、习题和上机练习等内容。

本书前 10 章主要介绍了：程序逻辑与程序设计语言、C 语言的基本概念、数据类型和运算符应用、三种基本结构的程序设计方法、数组、函数、指针、结构体与共用体、动态存储、编译预处理、位运算和文件等内容；第 11 章提供了一个综合应用实例。

本书可作为高等职业院校计算机及相关专业的教材，也可以作为“全国计算机等级考试二级 C 语言程序设计”的辅导教材，或作为自学 C 语言的参考用书。

图书在版编目（CIP）数据

C 语言程序设计教程 / 李敏主编. —北京：机械工业出版社，2007.1
（21 世纪高职高专规划教材系列）
ISBN 978-7-111-20742-9

Ⅰ. C… Ⅱ. 李… 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（2007）第 005199 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
策　　划：胡毓坚
责任编辑：车　忱
责任印制：李　妍

保定市中画美凯印刷有限公司印刷
2007 年 2 月第 1 版 · 第 1 次印刷
184mm×260mm · 16.25 印张 · 401 千字
0001－5000册
标准书号：ISBN 978-7-111-20742-9
定价：23.00元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639　88379641　88379643
编辑热线电话：（010）88379739

出 版 说 明

为了贯彻国务院发［2002］16 号文件《国务院关于大力推进职业教育改革与发展的决定》的精神，进一步落实《中华人民共和国职业教育法》和《中华人民共和国劳动法》，实施科教兴国战略，大力推进高等职业教育改革与发展，我们组织力量，对实现高等职业教育培养目标和保证基本教学规格的文化基础课程、专业技术基础课程和重点建设专业主干课程的教材进行了规划和编写。

本套教材内容涵盖了高职高专院校计算机类、电子信息类、通信类、自动化类、市场营销类专业的专业基础课、专业课以及选修课，为配合高职教育关于“培养 21 世纪与我国现代化建设要求相适应的一线科技实用型人才”的最新理念，我们特为本系列教材配备了实践指导丛书，以利于老师的教学和学生的学习。

本套教材将理论教学和实践教学紧密结合，图文并茂、内容实用、层次分明、讲解清晰，其中融入了作者长期的教学经验和丰富的实践经验。可作为各类高职高专院校的教材，也可作为各类培训班的教材。

机械工业出版社

前　言

程序设计人员在设计程序时，除了考虑数据结构和算法这两个因素，还应当考虑使用一种合适的语言。C 语言是一种应用十分广泛的计算机语言，其功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有高级语言的优势，又具有低级语言的许多特点，特别适合编写系统软件，已经成为计算机类本科生、高职高专生及中专生的必修课程。

本书符合教学大纲的基本要求，在编写过程中参考了《全国计算机等级考试二级考试大纲》中有关 C 语言程序设计的要求。在内容上突出以就业为导向，以实践技能为核心，倡导以学生为本位的教育理念，注重全面提高学生的职业实践能力和素养。在内容上力求准确精要、层次清晰、通俗易懂、实用性强，使学生在少走弯路的前提下对 C 语言产生浓厚的学习兴趣。

全书共分 11 章，前 10 章主要介绍了程序逻辑与程序设计语言、C 语言程序设计的基本概念、数据类型和运算符应用、三种基本结构的程序设计方法、数组、函数、指针、结构体与共用体、动态存储、编译预处理、位运算和文件等；第 11 章提供了一个综合实例，以便提高读者的综合编程能力和完成复杂程序设计的能力。在每章之后提供的习题和实训内容，突出了实用性，强调理论与实践相结合，培养了学生解决实际问题的能力。

本书不仅可以作为高职院校计算机及相关专业的教材，而且可以作为“全国计算机等级考试二级 C 语言程序设计”的辅导教材，或作为自学 C 语言的参考用书。

本书由李敏主编，全书各章节的编写分工如下：第 1 章的 1～2 节、第 5 章、第 6 章、第 11 章由李敏编写；第 1 章的 3～6 节、第 2 章由陈双编写；第 3 章、第 9 章、第 10 章由金梅编写；第 4 章、第 7 章、第 8 章由刘婷编写。

本书的电子教案和源代码可在 www.cmpbook.com 免费下载。

由于编写者水平有限，书中不足之处，请读者批评指正。

编　者

目　录

第1章　程序逻辑与程序设计语言

计算机是一种具有一定存储能力、在程序控制下自动工作的电子设备。为了使计算机发挥巨大作用，需要为它编写各类程序。编写程序时，不仅要认真考虑程序的数据结构和算法，而且还要考虑用一种语言来表示。

本章的主要内容包括：

- 程序与程序逻辑
- 程序设计语言
- C 语言概述
- 上机运行 C 程序的步骤与方法

1.1　程序与程序逻辑

1.1.1　程序

著名计算机科学家沃思（Niklaus Wirth）提出一个公式：程序=数据结构+算法。即一个程序应该包括两方面的内容：数据结构和算法。数据结构（data structure）是对数据的描述，在程序中要指定数据的类型和数据的组织形式。算法（algorithm）是对操作的描述，即操作步骤，是用来解决做什么和怎么做的问题。

任何一个程序都是为了解决特定的问题而编写的，因此编写程序前就必须先针对问题进行分析、规划和设计，然后利用工具来设计程序。所以，程序设计人员在设计程序时，除了考虑数据结构和算法这两个因素之外，还应当采用正确的程序设计方法进行程序设计，并且考虑用一种语言来表示。因此，一个程序应该表示为：

程序=数据结构+算法+程序设计方法+语言工具和环境。

一般认为，实现功能要求的具有一定上下文关系的程序中，包括代码的组织方式、算法和设计方法等方面的内容称为程序逻辑。通常情况下程序逻辑主要取决于程序设计的优劣，好的程序设计可以提供一个非常流畅的上下文关系。程序员所做的工作，其实就是组织好程序代码内部的逻辑关系。

1.1.2　算法

1．算法概述

算法是指为解决某个问题而采用的方法和步骤。比如，高职院校的学生要报考专升本，首先需要填写报名表，上交报名费用，领取准考证，然后按照规定的时间到指定的地点参加考试，得到录取通知书后，到指定的高等院校报到。这些步骤是按一定的顺序进行的，每个步骤不能缺少，它们的次序也不能颠倒。实际上，在日常生活中，人们做任何事情都有一定的方法和步骤，只是由于习惯问题，意识不到做每件事情都需要事先设计出行动步骤。可见，

算法体现了人们解决某一类问题时的思维方法和过程，描述了人类解决某类问题所依据的规则和操作。

计算机算法可分为两大类：数值运算算法和非数值运算算法。数值运算算法主要用于求解数值问题，如求函数值、求方程的根等。一般数值运算有现成的模型，可以运用数值分析方法，因此对数值运算算法的研究比较深入，各种数值运算都有比较成熟的算法可供选用。非数值运算算法常用于事务管理领域，如人事管理、行车调度管理等。由于非数值运算要求各异，很难规范化，因此一般只对一些典型的非数值运算算法作比较深入的研究。

下面通过一个简单算法举例，让读者了解如何设计一个算法。

【例 1-1】 求 1+2+3+4+5+6。

用最原始的算法表示：

步骤 1：先求解 1+2，得到结果 3。

步骤 2：将步骤 1 得到的和 3 再加 3，得到结果 6。

步骤 3：将步骤 2 得到的和 6 再加 4，得到结果 10。

步骤 4：将步骤 3 得到的和 10 再加 5，得到结果 15。

步骤 5：将 15 再加 6 得到最后的结果 21。

这种最原始的算法太繁琐，如果参与运算的对象很多，需要书写大量的步骤，所以这样表示算法是不可取的，需要用一种通用的表示方法。

设两个变量：一个变量 p 代表被加数，另一个变量 q 代表加数，并把每一步求出的和放在被加数变量 p 中，并采用循环算法来求解结果，现将算法改写如下：

步骤 1：使 p=1。

步骤 2：使 q=2。

步骤 3：使 p+q→p。

步骤 4：使 q 的值加 1，即 q+1→q。

步骤 5：如果 q 不大于 6，返回重新执行步骤 3 及其后的步骤；否则算法结束，最后得到的 p 值就是 1+2+3+4+5+6 的和。

上面算法中，步骤 3 到步骤 5 组成一个循环，在实现算法时，要多次执行循环，直到 q 值大于 6 不再返回步骤 3 为止。如果求解成千上万个数的和，由于计算机是高速进行运算的自动机器，实现循环是轻而易举的。可见上述算法具有通用性和灵活性，也是计算机能实现的较好的算法。

2. 算法的特性

- 有穷性：一个算法应包含有限的操作步骤，且每一步都可在有穷的时间内完成。
- 确定性：算法中每一个步骤必须有确切的含义，并且在任何条件下，算法只有惟一的一条执行路径，即对于相同的输入只能得出相同的输出。
- 可行性：一个算法是能行的，即算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。
- 有零个或多个输入：这些输入取自某个特定的对象的集合。
- 有一个或多个输出：这些输出是同输入有着某些特定关系的量，在一个完整的算法中至少会有一个输出。

3. 算法的表示

1966 年，Bohra 和 Jacopini 证明了任何单入口单出口没有死循环的程序都可以由三种基本的控制结构构造出来。这三种基本结构就是顺序结构、选择结构和循环结构，它们构成表示一个良好算法的基本单元。

顺序结构是按照一定的运算顺序，依次完成指定的运算功能。顺序结构的特点是：程序从入口开始，自上而下按顺序执行所有操作，直到出口为止。

选择结构表示程序的处理步骤出现了分支，它需要根据某一特定的条件选择其中的一个分支执行。选择结构有单选择、双选择和多选择三种形式。值得注意的是，对于双选择形式，在两个分支中只能选择一个且必须选择一个分支执行，但是不论选择了哪一条分支执行，最后流程都一定到达结构的出口处。

循环结构表示程序反复执行某个或某些操作，直到循环条件不成立时才可终止循环。循环结构的基本形式有两种：当型循环和直到型循环。

用这三种基本结构作为表示一个良好算法的基本单元，整个算法的结构是由上而下地将各个基本结构顺序排列起来，从而使算法质量得到保证和提高。

算法的表示方法很多，常用的有自然语言、传统流程图、N-S 流程图、伪代码、计算机语言等。

（1）用自然语言表示算法。自然语言就是人们日常使用的语言，可以是汉语、英语或其他语言。下面通过实例来说明用自然语言来描述三种基本结构的算法。

【例 1-2】 已知 a 的值是 7，b 的值是 10，将 a、b 的值互换，互换后 a 的值为 10，b 的值为 7，然后输出交换后 a、b 的值。

算法分析：因为 a、b 两个变量的值不能直接交换，所以，解决这一问题的关键是需要引入第 3 个变量。设第 3 个变量为 c，其交换步骤可以用自然语言描述如下：

步骤 1：把 7 赋给变量 a。

步骤 2：把 10 赋给变量 b。

步骤 3：将变量 a 的值赋给变量 c。

步骤 4：将变量 b 的值赋给变量 a。

步骤 5：将变量 c 的值赋给变量 b。

步骤 6：输出变量 a 和变量 b 的值。

步骤 7：算法结束。

【例 1-3】 输出 a、b 两个不同数中的大数。

算法分析：对输入的两个数进行判断，并将其中的大数输出。用自然语言描述如下：

步骤 1：输入 a 和 b 的值。

步骤 2：判断 a 大于 b 否，如果 a 大于 b，执行第 3 步，否则执行第 4 步。

步骤 3：输出 a 的值。

步骤 4：输出 b 的值。

步骤 5：算法结束。

【例 1-4】 求 1+2+3+…+100。

算法分析：如果一步一步地求和，需要写出 99 个步骤，若求 1+2+3+…+1000，则需要写出 999 个步骤，显然是不可取的，应当找一种通用的表示方法。这里设 p 为被加数，q 为

加数，并将每一步的和放在 p 中。用自然语言描述如下：

步骤 1：使 p=1。

步骤 2：使 q=2。

步骤 3：使 p+q，和仍放在 p 中，可表示为 p+q→p。

步骤 4：使 q 的值加 1，即 q+1→q。

步骤 5：如果 q 不大于 100，返回重新执行步骤 3、步骤 4 和步骤 5。否则，算法结束。最后得到 p 的值就是 1+2+3+…+100 的和。

由上述例子看出，一个算法由若干操作步骤构成，并且这些操作是按一定的控制结构所规定的次序执行的。例 1-2 中的 7 个操作步骤是自上而下顺序执行的，称之为顺序结构。例 1-3 中的操作步骤不是按操作步骤顺序执行，也不是所有步骤都执行。如第 3 步和第 4 步就不能同时被执行，它们需要根据条件判断决定执行哪个操作，这种结构称为分支结构。在例 1-4 中不仅包含了判断，而且需要重复执行。如第 3、第 4 步骤就需要根据条件判断是否重复执行，并且一直延续到条件“q 大于 100”为止，这种具有重复执行功能的结构称为循环结构。

使用自然语言表示算法通俗易懂，但文字冗长，容易出现歧义，特别是对于包含分支和循环的算法，更是如此。因此，除了很简单的问题，一般不用自然语言描述算法。

（2）用流程图表示算法。流程图是用一些图框表示各种操作。美国国家标准化协会 ANSI 规定了一些常用的流程图符号，已为世界各国程序工作者普遍采用。流程图符号如图 1-1 所示。下面分别用流程图表示三种基本结构的算法。

【例 1-5】 将例 1-2 的算法用流程图表示。流程图如图 1-2 所示。

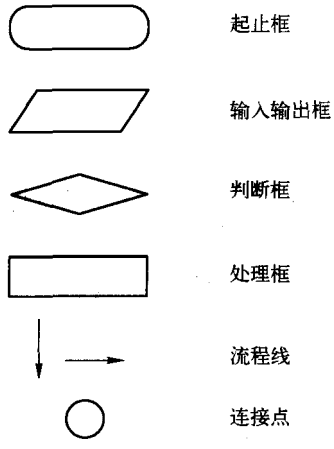

图 1-1 流程图符号

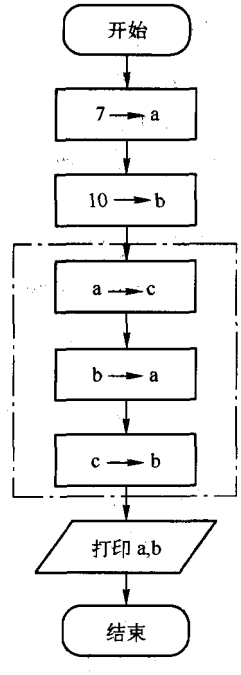

图 1-2 例 1-2 算法流程图

【例 1-6】　将例 1-3 的算法用流程图表示。流程图如图 1-3 所示。

【例 1-7】　将例 1-4 的算法用流程图表示。流程图如图 1-4 所示。

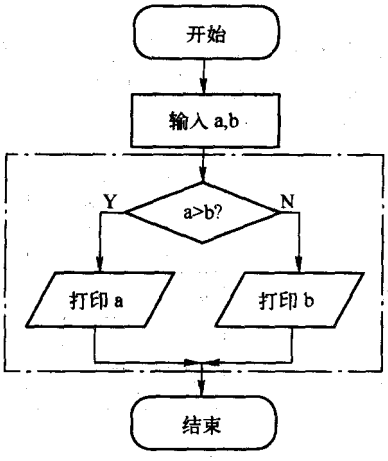

图 1-3　例 1-3 算法流程图

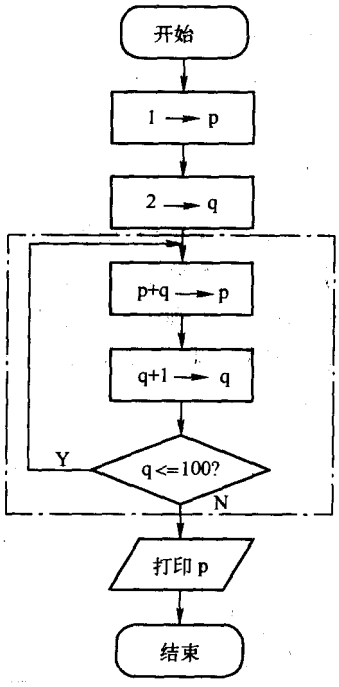

图 1-4　例 1-4 算法流程图

可见，三种基本结构的共同特点是：

- 只有一个入口。
- 只有一个出口。
- 结构内的每一部分都有机会执行。
- 结构内不存在“死循环”。

由以上三种基本结构顺序组成的算法结构，可以解决任何复杂的问题。可见，用流程图表示算法，直观、形象，易于理解，不会产生歧义。因为流程图便于交流，又特别适合初学者使用，所以对于一个程序设计工作者来说，会看会用流程图是必要的。

需要注意的是由于流程线是反映流程的执行先后次序的，不要忘记画箭头，否则就难以判定各框的执行次序。

（3）用 N-S 流程图表示算法。N-S 流程图是美国学者 I. Nassi 和 B. Shneiderman 于 1973 年提出的一种新的流程图。N-S 流程图的主要特点是取消了带箭头的流程线，全部算法写在一个矩形框内，在该框内还可以包含其他的从属于它的框。这种流程图适于结构化程序设计，因而很受欢迎。

N-S 流程图使用如图 1-5、图 1-6、图 1-7、图 1-8 所示的流程图符号。从图中可见，N-S 流程图如同一个多层的盒子，又称盒图。

图 1-5 表示顺序结构，它是由 A 和 B 两个框组成的。图 1-6 表示选择结构，当 p 条件成立时执行 A 操作，p 不成立执行 B 操作。图 1-7 和图 1-8 表示循环结构，其中图 1-7 表示当

型循环结构，图 1-8 表示直到型循环结构。

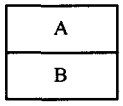

图 1-5 顺序结构

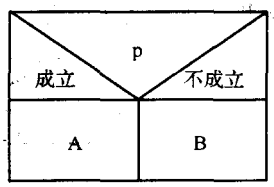

图 1-6 选择结构

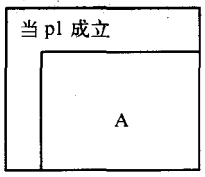

图 1-7 当型循环结构

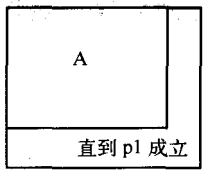

图 1-8 直到型循环结构

下面用 N-S 流程图分别表示例 1-2、例 1-3 和例 1-4 的算法。

【例 1-8】 将例 1-2 的算法用 N-S 流程图表示。如图 1-9 所示，该图与图 1-2 对应。

【例 1-9】 将例 1-3 的算法用 N-S 流程图表示。如图 1-10 所示，该图与图 1-3 对应。

【例 1-10】 将例 1-4 的算法用 N-S 流程图表示。如图 1-11 所示，该图与图 1-4 对应。

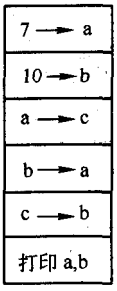

图 1-9 例 1-2 算法 N-S 图

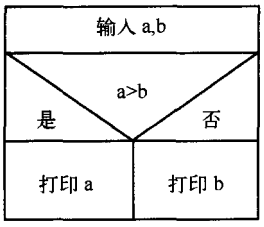

图 1-10 例 1-3 算法 N-S 图

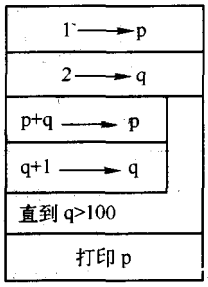

图 1-11 例 1-4 算法 N-S 图

可见，一个结构化的算法是由一些基本结构顺序组成的，在基本结构之间不存在向前或向后的跳转，流程的转移只存在于一个基本结构范围之内。

（4）用伪代码表示算法。伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。它的表示形式比较灵活自由，而且由于与计算机语言比较接近，因此可以方便地过渡到计算机程序。用伪代码写算法并无固定的、严格的语法规则，只要用清晰易读的形式把意思表达清楚即可。下面通过一个例子来说明。

【例 1-11】 将例 1-4 的算法用伪代码表示。

```
BEGIN（算法开始）
1→p
2→q
while q<=100
{
  p+q→p
  q+1→q
}
print p
END（算法结束）
```

用伪代码表示算法时，可以用英文伪代码，也可以用中文伪代码，还可以中英文混用，

由于篇幅所限，这里不再一一举例说明。

用伪代码很容易写出结构化的算法，书写格式比较自由，容易表达出设计者的思想。用伪代码写的算法很容易修改，而这却是用流程图表示算法时不便处理的。因此，流程图适宜表示一个算法，但设计算法时常用伪代码。

（5）用计算机语言表示算法。要完成一项工作，包括设计算法和实现算法两个部分。前面介绍的只是描述算法，要得到运算结果，就必须实现算法。实现算法的方法很多，这里要求用计算机实现算法。计算机无法识别流程图和伪代码，只能识别并执行用计算机语言编写的程序，因此，用流程图或伪代码描述出一个算法后，还要将它转换成计算机语言程序。

和伪代码不同的是，用计算机语言表示算法必须严格遵循所用语言的语法规则。下面通过例子将前面介绍过的算法用 C 语言表示。

【例 1-12】 将例 1-8 表示的算法（将变量 a,b 的值互置）用 C 语言表示。

```
#include <stdio.h>
main()
{
  int a,b,c;                    /*定义 a,b,c 为整型变量*/
  a=7;                          /*给 a 赋以整数 7*/
  b=10;                         /*给 b 赋以整数 10*/
  c=a;                          /*把 a 的值赋给 c*/
  a=b;
  b=c;
  printf("a=%d,b=%d\n",a,b);    /*以整型形式输出变量 a,b 的值*/
}
```

【例 1-13】 将例 1-9 表示的算法（输出 a,b 两个不同数中的大数）用 C 语言表示。

```
#include <stdio.h>
main()
{
  int a,b,c;
  scanf("%d%d",&a,&b);          /*格式输入函数，从键盘输入两个整型数分别给变量 a,b*/
  if(a>b) printf("%d\n",a);     /*如果 a 大于 b，输出 a 的值*/
  else printf("%d\n",b);        /*否则输出 b 的值*/
}
```

【例 1-14】 将例 1-10 表示的算法（求 1+2+3+…+100）用 C 语言表示。

```
#include <stdio.h>
main()
{
  int p,q;
  p=1;q=2;
  while(q<=100)       /*循环条件，只有 q 小于等于 100 时，执行下面的循环体*/
  {
    p=p+q;            /*将变量 p、q 之和赋给变量 p，实现累加*/
    q=q+1;            /*循环控制变量 q 增 1*/
```