

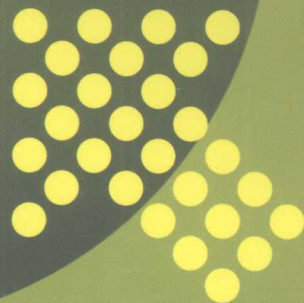
21世纪高等学校规划教材



C YUYAN CHENGXU·SHEJI

C语言程序设计

戴 锐 朱信树 罗 勇 主 编
洪微明 陈义新 王 瑞 主 编
沈锦明 参 编



中国电力出版社

<http://jc.cepp.com.cn>

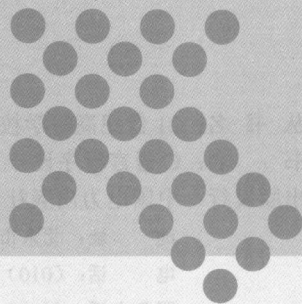
21世纪高等学校规划教材



C YUYAN CHENGXU SHEJI

C语言程序设计

主 编 罗 勇
副主编 戴 锐 朱信树 王 瑞
参 编 洪微明 陈义新 沈锦明



中国电力出版社

<http://jc.cepp.com.cn>

内容提要

本书为 21 世纪高等学校规划教材。

本书从 C 语言的基本概念出发,介绍了 C 语言程序设计的一般方法。全书共 12 章,内容包括 C 语言的基本概念、数据及运算、三大程序结构、数组、函数、指针、编译预处理、结构体和共用体、位运算、文件。书中每章都有典型程序实例和习题,并在附录中附有《全国计算机等级考试二级 C 语言程序设计考试大纲(2008 年版)》和《2008 年 4 月全国计算机等级考试二级 C 语言程序设计笔试试卷及参考答案》。

本书可作为高职高专院校计算机及相关专业的教材,也可作为备考等级考试和其他从事计算机编程人员的参考用书。

图书在版编目(CIP)数据

C 语言程序设计 / 罗勇主编. —北京:中国电力出版社, 2009

21 世纪高等学校规划教材

ISBN 978-7-5083-7991-3

I. C… II. 罗… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 198122 号

丛 书 名: 21 世纪高等学校规划教材

书 名: C 语言程序设计

出版发行: 中国电力出版社

地 址: 北京市三里河路 6 号

邮政编码: 100044

电 话: (010) 68362602

传 真: (010) 68316497, 88383619

服务电话: (010) 58383411

传 真: (010) 58383267

E-mail: infopower@cepp.com.cn

印 刷: 北京丰源印刷厂

开本尺寸: 184mm×260mm 印 张: 15.75 字 数: 350 千字

书 号: ISBN 978-7-5083-7991-3

版 次: 2009 年 1 月北京第 1 版

印 次: 2009 年 1 月第 1 次印刷

印 数: 0001—3000 册

定 价: 26.00 元

敬 告 读 者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

计算机技术的飞速发展,给社会进步带来了很强的发展动力。现在,计算机已广泛应用于社会生活的各个领域,成为当今社会不可缺少的工具。进入 21 世纪,计算机在各个领域的应用越来越广泛,与人们的联系也越来越紧密。由于计算机无法识别人类的语言,人们只能通过程序与之交流,要求其完成各种各样的工作。因此,掌握程序设计语言在当今社会显得尤为重要。

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。它适合作为系统描述语言,既可以用来编写系统软件,也可以用来编写应用软件。在众多的软件开发语言中,C 语言是软件开发人员常用的编程工具,在工程实践中得到了广泛应用,受到了众多程序设计者的欢迎。

随着计算机技术的飞速发展,为适应当前我国高职高专人才培养的发展方向,满足高职高专教育教学改革的需要,配合学生应对全国计算机等级考试,我们编写了此教材。

《C 语言程序设计》作为高职高专学生学习计算机编程的入门教材,我们在编写过程中参考了大量国内同类教材,吸收了这些教材的优点,同时又保持了自己的风格。本书着重介绍 C 语言程序设计的基本概念、设计思想、程序设计方法和基本技巧,使得学生通过学习,可掌握 C 语言程序设计的基本方法,为今后工作中解决一些软件方面的实际问题提供一种技术手段,并能结合社会生产实际进行应用程序的研制和开发。全书共 12 章,主要介绍 C 语言的基本概念、数据及运算、三大程序结构、数组、函数、指针、编译预处理、结构体和共用体、位运算、文件等。书中每章都有典型程序实例和习题,并在附录中附有《全国计算机等级考试二级 C 语言程序设计考试大纲(2008 年版)》和《2008 年 4 月全国计算机等级考试二级 C 语言程序设计笔试试卷及参考答案》。

本书适合作为高职高专院校计算机及相关专业的教材,也可作为备考等级考试和其他从事计算机编程人员的参考用书。

本书由罗勇主编和统稿。其中戴锐编写第 8 章、第 10 章和第 12 章;朱信树编写第 6 章和第 7 章;王瑞编写第 3 章和第 4 章;洪微明编写第 1 章和第 9 章,陈义新编写第 2 章;沈锦明编写第 5 章;罗勇编写第 11 章和附录。

在编写过程中,孙奕学、时军、卢银花和王懿华等对本书提供了许多宝贵意见,另外还得到了计算机信息系全体教师的大力支持,在此一并表示衷心感谢。

由于编者水平有限、时间仓促,书中难免有错误和不妥之处,恳请读者批评指正,并多多提出宝贵意见。

编 者

2008 年 10 月

目 录

前 言

第 1 章 C 语言概述	1
1.1 C 语言的发展与特点	1
1.2 C 语言程序结构	2
1.3 算法与程序设计	6
1.4 C 程序的上机步骤	9
习题	12
第 2 章 基本数据类型、运算符及表达式	14
2.1 C 语言的数据类型	14
2.2 标识符、常量和变量	15
2.3 整型数据	17
2.4 实型数据	19
2.5 字符型数据	20
2.6 算术运算符和表达式	23
2.7 赋值表达式	25
2.8 逗号运算符	27
习题	27
第 3 章 顺序结构程序设计	30
3.1 C 语句概述	30
3.2 赋值语句	30
3.3 数据输出	31
3.4 数据输入	38
3.5 空语句和复合语句	41
3.6 程序举例	42
习题	44
第 4 章 选择结构程序设计	48
4.1 关系运算符和关系表达式	48
4.2 逻辑运算符和逻辑表达式	49
4.3 if 语句	51
4.4 if 语句的嵌套	54
4.5 条件运算符	57
4.6 switch 语句	58

4.7 程序举例	62
习题	66
第 5 章 循环结构程序设计	71
5.1 while 语句和用 while 语句构成的循环结构	71
5.2 do-while 语句和用 do-while 语句构成的循环结构	74
5.3 for 语句和用 for 语句构成的循环结构	75
5.4 循环结构的嵌套	78
5.5 break 和 continue 语句在循环体中的作用	80
5.6 程序举例	81
习题	83
第 6 章 数组	88
6.1 一维数组的定义和引用	88
6.2 二维数组的定义和引用	92
6.3 字符数组和字符串	95
6.4 字符串处理函数	98
6.5 程序举例	100
习题	101
第 7 章 函数	104
7.1 函数定义的一般形式	104
7.2 函数的参数和函数的返回值	106
7.3 函数的调用	108
7.4 函数声明和函数原型	109
7.5 函数的嵌套调用	110
7.6 函数的递归调用	111
7.7 变量的作用域	112
7.8 变量的存储类型	115
7.9 内部函数和外部函数	119
习题	120
第 8 章 指针	125
8.1 地址、指针的基本概念和指针变量	125
8.2 指针变量的定义和指针运算	126
8.3 指针与数组	129
8.4 指针与字符串	139
8.5 指针与函数	141
8.6 指针操作的小结	146
习题	147

第 9 章 编译预处理	151
9.1 文件包含处理——#include	151
9.2 宏定义——#define	152
习题	155
第 10 章 结构体、共用体和用户自定义类型	158
10.1 结构体类型定义和结构体变量	158
10.2 结构体数组	162
10.3 结构体指针、动态存储分配和链表	164
10.4 共用体类型	172
10.5 枚举类型	175
10.6 用户自定义类型	177
习题	178
第 11 章 位运算	183
11.1 位运算符和位运算	183
11.2 位段	187
习题	190
第 12 章 文件	193
12.1 C 文件概述	193
12.2 文件的打开与关闭	194
12.3 文件的读写	197
12.4 文件的随机读写	204
习题	206
附录 A ASCII 码表	210
附录 B C 语言运算符的优先级和结合性	211
附录 C C 常用库函数	213
附录 D C 语言的关键字	219
附录 E 全国计算机等级考试二级 C 语言程序设计考试大纲 (2008 年版)	220
附录 F 2008 年 4 月全国计算机等级考试二级 C 语言程序设计 笔试试卷及参考答案	223
附录 G 习题参考答案	232
参考文献	242

第1章 C语言概述

计算机是人类 20 世纪最伟大、最重要的发明之一，它已广泛用于社会生活的各个领域，成为当今社会不可缺少的工具。进入 21 世纪以来，计算机在各个领域的应用越来越广泛，与人们的联系也越来越紧密。由于计算机无法识别人类的语言，人们只能通过程序与之交流，要求其完成各种各样的工作。

C 语言是当今世界上应用最广泛、影响最深远的高级程序设计语言之一。它适合作为系统描述语言，既可以用来编写系统软件，也可以用来编写应用软件。在众多的软件开发语言中，C 语言是软件开发人员最常用的编程工具，在工程实践中得到了广泛应用。C 语言是一种既具有一般高级语言特性，又具有低级语言特性的程序设计语言。它具有丰富灵活的数据结构、简洁而高效的语句表达、清晰的程序结构、良好的可移植性等特点，从而受到了众多程序设计者的喜爱。

本章主要介绍 C 语言的发展与特点、C 语言程序结构、算法与程序设计，以及 C 语言程序的上机步骤。

1.1 C语言的发展与特点

1.1.1 C语言的发展概况

C 语言是一种得到广泛应用并受到众多用户欢迎的计算机语言，它既可作为系统软件的描述语言，也可用来开发应用软件。20 世纪 60 年代，贝尔实验室的 Thompson 开始开发 Unix 操作系统，由于无法忍受汇编语言的烦琐，他采用 BCPL 语言，并把它改造成更适合编写 Unix 的 B 语言。

1971 年，Ritchie 开始和 Thompson 合作开发 Unix，当时的工作主要是改造 B 语言，使其更加成熟，后来这个语言被命名为 C 语言。1973 年 Thompson 和 Ritchie 用 C 语言重写了 Unix。值得一提的是在 1983 年，Thompson 和 Ritchie 共同获得了世界计算机科学技术界的最高奖——图灵奖，虽然这个奖的授予是因为他们在 Unix 上所做的开创性工作，但谁也不能忽视 C 语言在其中发挥的巨大作用。

C 语言自身也在不断完善，并在很多计算机应用领域大显身手。人们发现 C 语言不仅仅局限于用来编写系统软件，还更适合用来编写应用软件。1989 年美国国家标准协会 (ANSI) 发布了一个完整的 C 语言标准，人们习惯称其为 ANSI C。ANSI C 在 1990 年被国际标准化组织 (ISO) 全盘采纳，并在 1999 年做了一些必要的修正和完善后，ISO 又发布了新的 C 语言标准，简称为 C99。

由于网络和多媒体等技术发展很快, 计算机操作系统平台也在不断更新, 目前 C 语言逐渐偏居于一隅, 取而代之的是 C++、Java 和 C# 等较为时尚的语言。那么, 在此情况下还有必要学习 C 语言吗? 其实 C 语言可以说是 C++、Java 和 C# 等语言的基础, 由它引出了不少后继语言, 还有许多新语言从 C 语言中汲取营养, 吸收了它的不少优点。所以说, 学好 C 语言对以后学习其他语言大有帮助。

目前在微型计算机上使用的有 Microsoft C、Quick C 和 Turbo C 等多种版本, 这些 C 语言版本大同小异。本书将在 Microsoft Visual C++ 6.0 的环境下对 C 语言进行讲解。

1.1.2 C 语言的特点

C 语言之所以发展迅速, 而且成为最受欢迎的语言之一, 主要是因为它具有强大的功能。许多著名的系统软件, 如 Unix/Linux、Windows、DBASE III PLUS、DBASE IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序, 就更能显示 C 语言的优势, 像 PC-DOS、WORDSTAR 等就是用这种方法编写的。概括起来, C 语言主要有以下特点:

(1) C 语言是一种结构化程序设计语言。C 程序由一系列函数组成, 函数之间可以相互调用, 程序的层次结构较为清晰。

(2) C 语言具有较好的移植性。由于 C 语言有统一的标准, 在处理输入、输出等问题上采用独立的库函数进行处理, 因此便于在不同的机器之间移植。

(3) C 语言具有类型丰富和使用灵活的运算符。这些运算符不仅具有一般高级语言所具有的算术运算和逻辑运算功能, 而且还具有位运算和复合运算的功能。

(4) C 语言除了能实现高级语言设计功能之外, 还可以对内存地址直接操作, 从而可部分替代汇编语言的功能。这一特点使得 C 语言成为编制系统软件的首选语言。

(5) 语法限制不太严格, 程序设计自由度大。“限制”和“灵活”是一对矛盾, 限制严格, 就易失去灵活性; 而强调灵活性, 就必然放松限制。从这个角度来看, 使用 C 语言编程, 对编程者程序设计技巧的要求更高一些。

(6) 生成的目标代码质量高, 运行速度较快。

C 语言的优点很多, 但是它也存在一些缺点, 如运算符优先级太多, 语法定义不太严格等, 此外, 相对而言 C 语言较难学习和掌握。尽管如此, 由于它具有上述诸多优点, 仍然是一种颇受欢迎而且较为实用的程序设计语言, 在系统开发、科学计算、自动控制等各个领域被广泛应用。

1.2 C 语言程序结构

1.2.1 简单的 C 语言程序介绍

在学习 C 语言的具体语法之前, 先通过一些简单的 C 程序示例来初步了解 C 语言的基本结构的组成。

【例 1.1】 在屏幕上显示一行欢迎文字。

程序如下:

```
#include <stdio.h>           /*编译预处理命令*/
main( )                     /*主函数*/
{
    printf ("Welcome to C world!\n");
}
```

程序运行结果如下:

Welcome to C world!

说明:

(1) C 语言程序是由函数组成的, 每一个函数完成一个相对独立的功能。在本例中, 有一个名为 main 的函数, 它又称为主函数, 起着控制程序执行的作用。一个完整的 C 程序有且仅有一个 main 函数。

(2) printf 是一个 C 语言提供的标准输出函数, 它的作用是在屏幕上输出指定的内容, 本例输出 “Welcome to C world!” 字符串。字符串末尾的 \n 是转义字符, 其作用是回车换行, 即光标自动跳到下一行最左端。

(3) stdio.h 是一个头文件, 里面包含 printf 库函数的一些必要的说明信息。#include 是编译预处理命令, 它负责把一对尖括号内的头文件包含在程序内。

(4) /*和*/之间的文字是注释部分, 起着对程序解释说明的作用, 在程序编译时会自动忽略。编程者应养成书写程序注释的习惯, 以增强程序的可读性。

【例 1.2】输入两个整数, 计算它们的和, 并输出结果。

程序如下:

```
#include <stdio.h>           /*编译预处理命令*/
main( )                     /*主函数*/
{ int a, b, sum;
    scanf("%d%d",&a, &b);
    sum=a+b;                /*计算两数之和并存储到变量 sum 中*/
    printf("sum is %d\n",sum);
}
```

程序运行结果如下:

```
3 5<回车>           (从键盘输入两个整数)
sum is 8
```

说明:

(1) int a, b, sum; 语句声明了三个整型变量, 分别是 a、b 和 sum。

(2) scanf 函数是 C 语言提供的标准输入函数, 它的作用是从键盘输入所需的一些数据, 并赋值给相关的变量。scanf 函数中的 %d%d 表示输入两个数据分别给两个整型变量, &a,&b 表示变量 a 和变量 b 的地址。整条语句的作用是调用 scanf 函数, 从键盘输入两个数据, 分别送到地址 &a 和 &b 所指出的存储单元中。

(3) sum=a+b; 是赋值语句, 它的作用是取出变量 a 和变量 b 的值, 相加之后把求得

之和存入变量 c 中。

(4) 程序调用库函数 `printf` 用来输出结果, 首先输出一串普通字符 “sum is”, `%d` 说明这里将要输出的是一个整型的值, 它与后面的参数 c 对应, 表示输出的是变量 c 的值 “8”。

【例 1.3】 输入两个整数, 编写函数计算它们的乘积, 并输出结果。

程序如下:

```
#include <stdio.h>                /*编译预处理命令*/
main( )                          /*主函数*/
{ int pro(int x,int y);          /*函数声明*/
  int a,b,c;
  scanf("%d%d",&a, &b);
  c=pro(a,b);                    /*函数调用*/
  printf("result is %d\n",c);
}
int pro(int x,int y)             /*函数定义*/
{ int z;
  z=x*y;
  return(z);
}
```

程序运行结果如下:

```
3 5<回车>
result is 15
```

说明:

(1) C 语言的函数包括系统提供的库函数和自定义函数。用库函数实现程序固然是程序设计的一条捷径, 但是库函数毕竟不能完成所有的功能。当在函数库中找不到实现所需功能的库函数时, 就需要自己来编写定义函数。本程序中的 `pro` 函数便是一个自定义函数。

(2) `pro` 函数的圆括号内有两个整型变量 x 和 y , 叫做形式参数 (简称形参), 它的作用是接收从 `main` 函数传来的乘数和被乘数。在 `pro` 函数的函数体内计算两者之积, 然后使用 `return` 语句将计算结果返回给 `main` 函数。

(3) `main` 函数需要计算 a 和 b 的积时, 就调用 `pro` 函数来实现求积功能。调用时把变量 a 和 b 作为实在参数 (简称实参) 传递给 `pro` 函数的形参 x 和 y , 即 a 的值赋给形参 x , b 的值赋给形参 y , 因此 x 和 y 的积就是 a 和 b 的积。

(4) 被调用的函数既可以是系统提供的库函数 (如 `printf` 和 `scanf` 函数), 也可以是用户根据需要自己定义的函数 (如 `pro` 函数)。函数是 C 程序的基本单位, 用来实现特定的功能, 程序的全部工作都是由函数来完成。

1.2.2 C 语言程序结构

C 程序结构的一般形式如下:

```
编译预处理命令
全局变量定义
main( )
```

```

{
    变量定义序列
    语句序列
}
sub1( )                /*自定义函数 sub1*/
{
    变量定义序列
    语句序列
}
:
subn( )                /*自定义函数 subn */
{
    变量定义序列
    语句序列
}

```

通过前面我们所讲的例子和 C 语言程序一般的组成形式，我们对 C 语言程序结构有了一定了解：

(1) C 语言程序由函数构成。一个 C 程序至少包含一个主函数 **main**，任何 C 程序都是从 **main** 函数开始执行。C 程序可以包含一个主函数和若干个其他函数。函数是 C 程序的基本单位。C 语言的库函数非常丰富，标准 C 提供 100 多个库函数。C 语言的这种特点为模块化的程序设计和实现提供了方便。

(2) 一个函数由函数首部和函数体两个部分组成。

1) 函数首部，即函数的第一行，包括函数类型、函数名、函数参数（参数类型，参数名）。[例 1.3] 中的函数 **pro** 的首部如下：

int	pro	(int	x,	int	y)
↓	↓	↓	↓	↓	↓
函数类型	函数名	形参 x 的类型	形参名	形参 y 的类型	形参名

2) 函数体，即函数首部下面的大括号 { } 括起来的部分。函数体一般包含两个部分，即“声明部分”和“执行部分”。如 [例 1.3] 中的 **pro** 函数：

```

int pro(int x,int y)
{ int z;                /*变量声明部分*/
  z=x*y;                /*执行部分*/
  return(z);
}

```

①变量声明部分：用来定义程序中所用到的变量，如例 1.3 中 **main** 函数中的 **int a,b,c;** 和 **pro** 函数中的 **int z;** 当然，有的函数可以没有变量声明部分，如 [例 1.1] 中的 **main** 函数。

②执行部分：这部分由若干语句组成，程序中由这些语句向计算机系统发出操作命令。

(3) 函数在程序中的位置没有限制，但程序运行总是从主函数开始直到主函数结束。

(4) C 程序的书写格式自由，一行内可以写一个或几个语句，一个语句也可以分写在多个行上。C 程序没有行号，每个语句和变量定义必须用一个分号结尾。

(5) 可以用 **/* ... */** 对程序的任何部分进行注释，增加程序的可读性。

1.3 算法与程序设计

1.3.1 算法的概念

著名的计算机科学家沃思 (N.Wirth) 曾提出过一个经典的公式：程序 = 数据结构 + 算法。这个公式的重要性在于它说明了程序与数据结构以及算法的关系。

所谓算法，就是一个有穷规则的集合，它确定了一个解决某个特定类型问题的运算序列。简单地说，算法就是指为解决某个特定问题而采取的确定且有限的步骤。编写一个程序的关键就在于合理地组织数据和设计算法。解决一个问题可能会有多种算法，可以用不同的算法以及不同的程序设计语言来解决同一个问题，只是速度和效率上有所不同而已。

一个算法应当具有以下五个特性：

(1) 有穷性。一个算法包含的操作步骤应该是有限的。也就是说，在执行若干个操作步骤之后，算法将结束，而且每一步都在合理的时间内完成。

(2) 确定性。算法中的每一条指令必须有确切的含义，不允许有歧义。

(3) 可行性。算法中指定的操作，都可以通过已经验证过可以实现的基本运算执行有限次后实现。

(4) 有零个或多个输入。在计算机上实现的算法是用来处理数据对象的，在大多数情况下这些数据对象需要通过输入来得到。

(5) 有一个或多个输出。算法目的是为了求“解”，这些“解”只有通过输出才能得到。

1.3.2 算法的描述方法

算法可以用各种描述方法来进行描述，最常用的是伪代码和流程图。

伪代码是介于自然语言和计算机语言之间的一种代码，是帮助程序员指定算法的智能化语言。它不能在计算机上直接执行，但是使用起来比较灵活，无固定格式和规范，只要写出来自己和别人都能看懂即可。

流程图也是描述算法的很好的工具，传统的流程图由图 1.1 中所示的几种基本图形组成。



图 1.1 传统流程图的基本图形

由这些基本图形中的框和流程线组成的流程图来表示算法，形象直观，简单方便。但是，这种流程图对于流程线的走向没有任何限制，可以任意转向，在描述复杂的算法时所占篇幅较多，费时费力且不易阅读。

为了限制传统流程图的随意性,1973年,I.Nassi和B.Shneiderman提出用一种新的流程图来代替传统的程序流程图。这种流程图完全去掉了流程线,算法的每一步都用一个矩形框来描述,把一个个矩形框按执行的次序连接起来就是一个完整的算法描述。这种流程图用两位学者名字的第一个英文字母命名,称为N-S流程图(又称为盒状图)。

1.3.3 结构化程序设计

结构化程序设计由三种基本结构组成,用流程图表示的这三种基本结构如下:

(1) 顺序结构。顺序结构是最简单的基本结构。在顺序结构中,要求按顺序地执行按先后顺序排位的每一个最基本的处理单位。如先执行处理A,再执行处理B,图1.2(a)所示的是用传统流程图表示的顺序结构,图1.2(b)所示的是用N-S流程图表示的顺序结构。

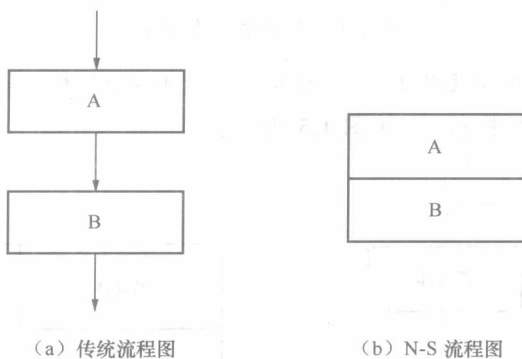


图 1.2 顺序结构流程图

(2) 选择结构。在选择结构中,要根据逻辑条件的成立与否,分别选择不同的处理。如图1.3所示,当逻辑条件成立时,执行处理A,否则执行处理B。

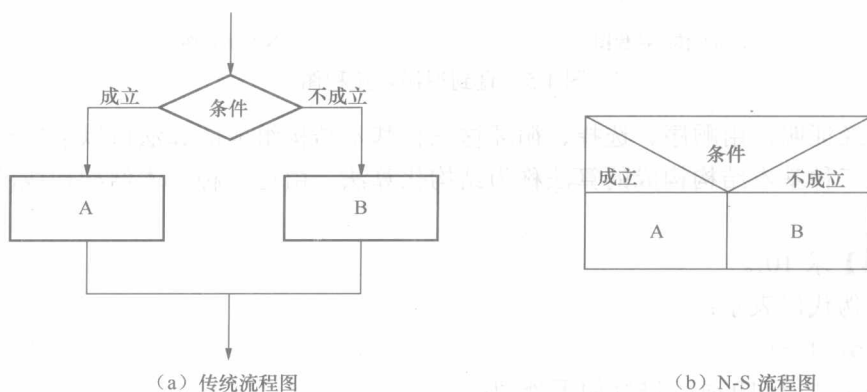


图 1.3 选择结构流程图

(3) 循环结构。循环结构分为当型循环和直到型循环。

1) 当型循环。在当型循环结构中,当逻辑条件成立时,就反复执行处理循环体,直到逻辑条件不成立时结束,如图1.4所示。

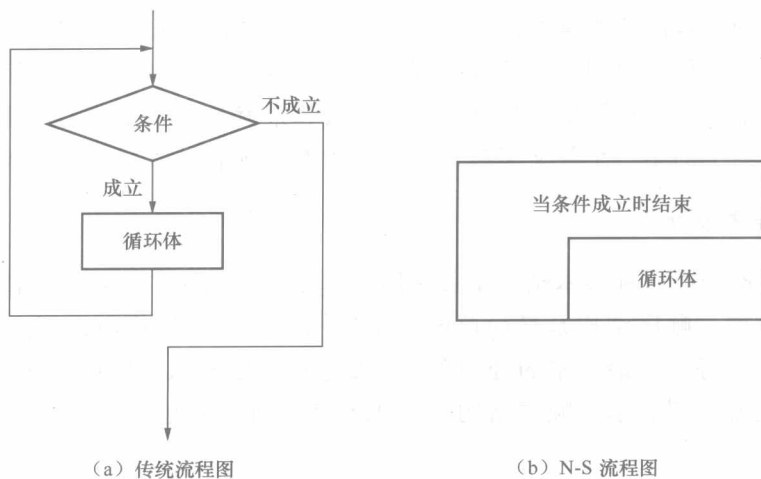


图 1.4 当型循环流程图

2) 直到型循环。在直到型循环中, 反复执行处理循环体, 直到逻辑条件成立时结束 (即逻辑条件不成立时继续执行), 如图 1.5 所示。

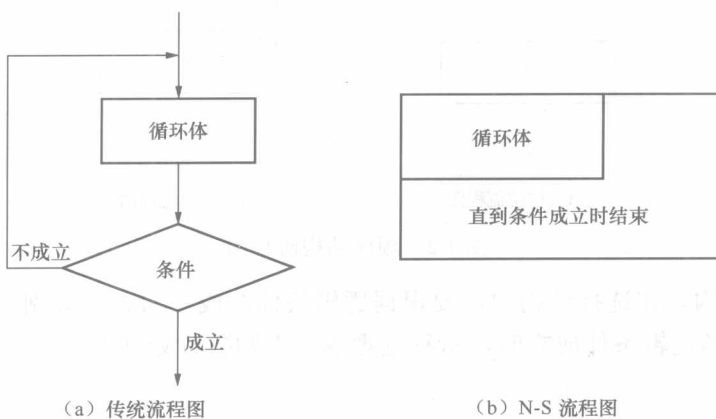


图 1.5 直到型循环流程图

理论已经证明, 由顺序、选择、循环这三种基本结构组成的算法可以解决任何复杂的问题。由这三种基本结构构成的算法称为结构化算法; 由这三种基本结构构成的程序称为结构化程序。

【例 1.4】求 $10!$ 。

(1) 用伪代码表示:

1) $1 \rightarrow p, 1 \rightarrow i$

2) 当 i 不大于 10 时, 进行如下处理:

① $p \times i \rightarrow p$

② $i+1 \rightarrow i$

3) $\text{printf}("%d", p);$

(2) 用传统流程图表示 (如图 1.6 所示)。

(3) 用 N-S 流程图表示 (如图 1.7 所示)。

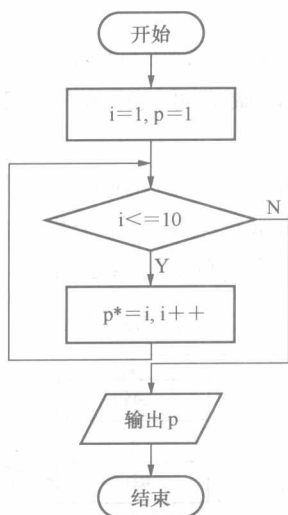


图 1.6 用传统流程图描述求 10!

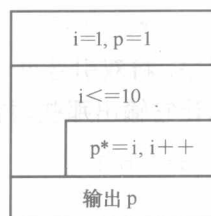


图 1.7 用 N-S 流程图描述求 10!

1.4 C 程序的上机步骤

1.4.1 C 程序的编辑、编译与运行

C 语言是高级程序语言，用它写出的程序通常称作 C 语言源程序（其扩展名为“.c”），人们容易使用、书写和阅读，但计算机却不能直接执行，因为计算机只能识别和执行特定二进制形式的机器语言程序。为使计算机能完成某个 C 语言源程序所描述的工作，就必须首先把这个源程序转换成二进制形式的机器语言程序，这种转换称为“C 程序的加工”。“C 程序的加工”包括“编译程序”、“连接程序”等，如图 1.8 所示。

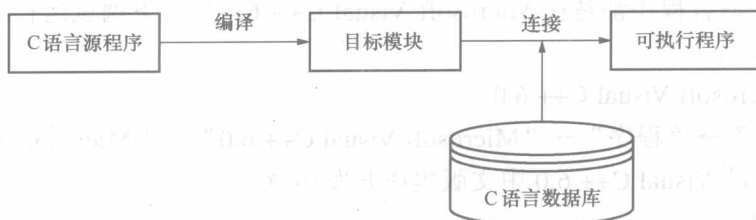


图 1.8 C 源程序加工过程

C 程序加工通常分两步完成：

第一步，由编译程序对源程序文件进行分析和处理，生成相应的机器语言目标模块，由目标模块构成的代码文件称为目标文件（其扩展名为“.obj”）。目标文件计算机还不能执行，因为其中缺少 C 程序运行所需要的一个公共部分——C 程序的运行系统。此外，一

般 C 程序里都要使用函数库提供的某些功能。例如前面例子用到标准函数库的一个输出函数（printf 是该函数的名字）。

第二步，加工和连接。这一工作由连接程序完成，将编译得到的目标模块与其他必要部分（运行系统、函数库提供的功能模块等）拼装起来，做成可执行程序（其扩展名为“.exe”）。图 1.8 说明了 C 程序的基本加工过程。

对前面简单 C 程序 [例 1.1] 进行加工后，就能得到一个与之对应的、可以在计算机上执行的程序。启动运行这个可执行程序，将能看到它的执行效果。这个程序的执行将得到一行输出，通常显示在计算机屏幕上，或者图形用户界面上的特定窗口里：

```
Welcome to C world!
```

如果修改程序，将双引号里一串字符换成其他内容，就可以让它输出那些内容。例如：

```
#include <stdio.h>
main( )
{
    printf ("This is a C program\n ");
}
```

这一程序加工后执行，就会输出：

```
This is a C program
```

C 程序加工过程的启动方式由具体 C 语言系统确定。

1.4.2 C 程序的上机步骤

C 语言程序的上机步骤如图 1.9 所示。

1.4.3 在 Microsoft Visual C++ 中编程

本书中的 C 语言程序都是在 Microsoft Visual C++ 6.0 环境下调试运行的，Visual C++ 6.0 的使用步骤如下：

1. 启动 Microsoft Visual C++ 6.0

单击“开始”→“程序”→“Microsoft Visual C++ 6.0”→“Microsoft Visual C++ 6.0”命令，就可以启动 Visual C++ 6.0 中文版集成开发环境。

2. 创建文件

单击主菜单中的“文件”→“新建”命令，在“新建”对话框中选择“文件”选项卡。在左边列出的选项中，选择“C++ Source File”；在右边相应的文本框中输入文件名称（假设为 program.c）及保存位置（假设为 D:\VC）；单击“确定”按钮，即进入 Microsoft Visual C++ 集成环境的代码编辑窗口，如图 1.10 所示。

3. 编辑代码

在 Microsoft Visual C++ 代码编辑窗口中，输入要编辑的 C 语言的源代码。

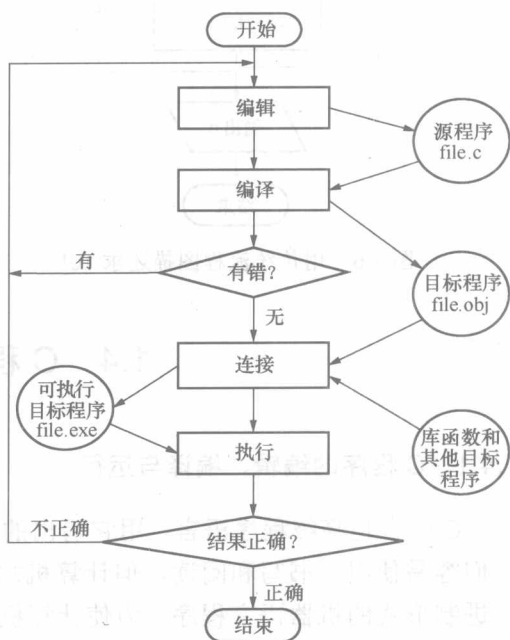


图 1.9 C 语言程序上机步骤