普通高等教育“十三五”规划教材

C CHENGXU SHEJI JIAOCHENG

C程序设计教程

（第二版）

胡桂珍　王　萱　杨华莉　易庆萍　编著

西南交通大学出版社

普通高等教育“十三五”规划教材

C 程序设计教程

（第二版）

胡桂珍　王　萱　杨华莉　易庆萍　编著

西南交通大学出版社
·成　都·

内 容 简 介

本书围绕C语言核心内容展开，在介绍核心语法的基础上，以培养动手编程能力为首要目标，注重基础知识，强调实践能力的培养。内容包括：C语言概述、基本数据类型、运算符、表达式、基本数据类型的输入输出、流程控制、函数、数组、指针、结构体、共用体、位运算和文件等。通过大量的典型程序，详尽地阐述了相关编程思想、方法、语法、算法、技巧和调试技术，激发学习者的兴趣，加深对相关知识的理解和掌握。重点和难点突出，力求讲清讲透，目的是让学习者能够学以致用，编写出自己满意的程序。在每章后精心设计了习题和同步实验，让学习者多多接触实际开发过程，提高编程能力和上机调试的能力，为进一步学习程序设计打下良好的基础。

本书可作为高等学校各专业学习C语言的教材，也适合作为程序设计的初学者或者有一定基础的学习者的自学教材。另外，本书配有教学所需的电子教案，提供所有例题的程序源代码、教学大纲等。您可以到西南交通大学出版社的网站免费下载。

图书在版编目（CIP）数据

C程序设计教程 /胡桂珍等编著. —2版. —成都：西南交通大学出版社，2015.2
普通高等教育“十三五”规划教材
ISBN 978-7-5643-3796-4

Ⅰ.①C… Ⅱ.①胡… Ⅲ.①C语言－程序设计－高等学校－教材 Ⅳ.①TP312

中国版本图书馆CIP数据核字（2015）第037505号

普通高等教育“十三五”规划教材

C程序设计教程

（第二版）

胡桂珍　王　萱　杨华莉　易庆萍　编著

*

责任编辑　孟苏成
特邀编辑　黄庆斌
封面设计　何东琳设计工作室
西南交通大学出版社出版发行
（四川省成都市金牛区交大路146号　邮政编码：610031　发行部电话：028-87600564）
http://www.xnjdcbs.com
四川森林印务有限责任公司印刷

*

成品尺寸：185 mm×260 mm　　印张：20.75
字数：520千
2015年2月第2版　　2015年2月第3次
ISBN 978-7-5643-3796-4
定价：36.00元

课件咨询电话：028-87600533
图书如有印装质量问题　本社负责退换

第一版前言

C 语言诞生已有几十年，一直保持着旺盛的生命力。C 语言标准也历经多次修改，从传统 C 发展为目前最新的标准 C99。C 语言在进一步完善，进一步发展壮大。C 语言的生命力不可能在未来 10 年、20 年内枯竭，虽然有 C++、Java 这样的后继者，但是到目前为止，它们依然没有取代 C 语言的迹象，精通 C 语言的精英们依然是各大软件企业高薪猎取的对象。

C 语言是一种短小精悍的计算机高级程序设计语言，它既具有高级语言的功能，又具有低级语言的功能，是一种可以用于开发系统软件和应用软件的程序设计语言，根据结构化程序设计原则设计并实现。C 语言具有丰富的数据类型，为结构化程序设计提供了各种数据结构和控制结构，能够实现汇编语言中的大部分功能。同时，用 C 语言编写的程序具有良好的可移植性。C 语言作为一种基础语言，从实用性和现实性两方面来看，学习和掌握它都是十分有必要的。

本书作者都是大学教师，一直工作在高等学校教学一线，承担“C 语言程序设计”课程的教学任务，有着丰富的教学经验，并长期从事 C 语言编程工作，并有着将自己认识、理解的“C 语言程序设计”介绍给大家的强烈愿望。在长期的教学实践中，作者认为弄懂基本的、主要的、核心的内容才是学习的重点，教材也应该围绕核心内容组织。本书就是采取了围绕核心内容展开的组织形式，在介绍核心语法的基础上，以培养动手编程能力为首要目标，把那些烦琐复杂的内容留待以后慢慢研究。另外，本书特别强调实践能力的培养，学习者首先应该学会用适当的开发工具编写、调试哪怕是最基本的程序，这是最主要的。“能够动手编写程序、调试程序”，就能激发学习者的信心和热情。学习者在编程实践中不断遇到问题、不断解决问题，自然就会明白许多细节，编写出自己满意的程序。

另外，C 语言标准是一种通用的行业标准，除了基本语法以外，标准库函数也是 C 语言中非常重要的一个组成部分。因此，C 语言教与学的重点还需要放在引导学习者熟悉标准库函数上面。

本书作者根据多年的教学经验和应用 C 语言的体会，将 C 语言的核心内容进行了合理组织，力求做到条理清晰、深入浅出，同时设计和精选了大量的例题和习题。本书的主要特点可归纳如下：

（1）按照循序渐进的原则，逐步引出 C 语言的概念。

（2）在文字叙述上力求条理清晰、简洁，以利于读者阅读。

（3）在讲解 C 语言的基本概念时，除了阐述理论外，还通过典型例题，着重强调了基本概念在程序设计中的应用，以利于读者阅读和理解。

（4）本书的重点是 C 语言的使用，体系结构是针对初学者的特点精心安排的，书中没有深奥的理论和算法，针对在例题中出现的每一种算法，都给出了详细的解释。

（5）本书包含了丰富的程序例题，但所举的例题都是易于理解的，并不涉及太多的硬件知识。通过对例题的阅读和分析，可以使读者更加全面地了解 C 语言的知识。

（6）每章的最后都附有一定量的习题，做这些习题对于读者巩固已学习的内容大有益处。

（7）学习编程离不开实验，在每章后都安排了以基本算法、综合编程为核心内容的实验，共15个，与课堂教学同步，提高学习者的实际编程能力。

（8）主张采用实际工程开发环境进行教学，让学习者接触实际开发工具，为今后工作打下良好的基础。

本书介绍标准C语言，符合ANSI/ISO C标准，所有程序都可以在实际工程环境中调试运行。为了尝试新的开发工具，数组和指针这两章的程序采用了Visual Studio 2008开发环境，其他章节的程序仍用Visual C++ 6.0开发环境，所提供的例题程序全部调试通过，用截图形式给出程序运行结果。使用的操作系统包括Windows XP、Windows7。学习本书的特别建议：不要过快陷入语法的细节。作者觉得，告诉学习者程序的主体，引导学习者编程，并在编程中激发自主学习意识、激发自主学习观念才是最重要的。另外，在学习时，除了教材以外，还得多看几本参考书籍，在比较中学习，可以从不同的侧面掌握相关学习内容。

如果在教学中发现与本教材有关的任何问题都可以与作者联系：gshglh@126.com。作者将尽力满足各位教师朋友的要求。本书将提供所有例题程序源代码，以及教学所需的ppt、教学大纲等。您可以到西南交通大学出版社的网站下载。

全书由靳桅、廖革元两位老师主审，得到了西南交通大学出版社和西南交通大学峨眉校区计算机与通信工程系各位同仁的大力支持和帮助，在此表示衷心感谢。特别感谢肖波老师，正是因为他的热心帮助，本书才得以按时出版。

书末参考文献所列书籍的内容、良好的风格和组织结构思想对作者产生了重要影响，因此本书也引用了这些书籍的部分实例程序。在此衷心感谢这些书籍的作者。在本书的编写过程中还参考了许多同行的著作，有的甚至还不方便列在参考文献目录中，作者在此一并表达感谢之情。

尽管作者尽了最大努力，也有良好且负责任的态度，但是由于学识所限，书中定有不完善之处，竭诚希望得到同行和读者的批评指正。

编　者

2010年11月20日

目　录

第 1 章　C 语言概述

学习目标

◆了解 C 语言的发展及主要特点
◆掌握 C 程序的基本结构
◆掌握算法的概念、特点及描述方法
◆掌握使用 Visual C++ 6.0 开发 C 程序的方法

1.1　C 语言简介

1.1.1　程序设计语言的发展

C 语言以其良好的可移植性和广泛的应用性深受中外广大用户的欢迎，成为当今使用最为广泛的计算机语言之一。它不但可以编写系统软件，还可以编写应用程序。

早期的系统软件主要由汇编语言编写而成。由于汇编语言过分依赖于计算机的硬件，致使程序的可读性和可移植性都很差。如果采用高级语言来编写系统软件，虽然可以提高程序的可读性和可移植性，但是高级语言又无法实现某些汇编的功能，比如对内存地址的操作、位操作等。于是，人们试图开发出一种既具有高级语言的特点，又具有汇编语言特性的语言来，C 语言就是这样一种语言。

C 语言的出现是与UNIX 操作系统联系在一起的，它的雏形是 1963 年由英国剑桥大学研制的一种高级编程语言 ——CPL（Combined Programming Language）语言。该语言较早期的 ALGOL 60 更接近硬件，但是规模较大，难以开发系统软件。20 世纪 60 年代末期剑桥大学的研究员 Martin Richards 在 CPL 的基础上进行改进，将其简化，研制出了 BCPL（Basic Combined Programming Language）语言。

1970 年，美国贝尔实验室的 Ken Thompson 对 BCPL 进一步优化，使该语言更为精炼，并用 BCPL 的第一个字母 B 为其命名，称为 B 语言。它只有一种数据类型，语言的功能也有限。1972 年，贝尔实验室的 Dennis Ritchie 以 B 语言为基础，引入了多种数据类型，研制出一种新的语言。它不但保持了 BCPL 和 B 语言的精练、接近硬件的优点，还克服了它们数据类型较少甚至没有、过于简单的缺点。给这个语言命名时，选用了 BCPL 的第二个字母 C 作为它的名字。C 语言就此诞生，并为 DEC PDP-11 计算机编写了第一个 UNIX 操作系统。

在 20 世纪 70 年代中期，C 语言逐渐代替了人们熟悉的其他语言，成为贝尔实验室的“官方”语言，其后声名远播，使用者也日益增多，而 Fortran、PL/1、APL、Pascal 等语言却逐渐被程序员们疏远，这一点就连 Dennis Ritchie 本人也倍感吃惊。1980 年以前已有多种 C 编

译程序在市场上出现。不仅是在非 UNIX 系统的机器上，就连单片机这样小的机器上也出现了一些 C 语言编译程序。

1978 年，Brian Kernighan 和 Dennis Ritchie（K&R）合作出版了 *The C Programming Language*，由它产生了 C 语言版本的基础，即标准 C。1983 年，ANSI（美国国家标准化协会）在标准 C 的基础上做了扩充和发展，制定出新的标准，称为 ANSI C。

在经过 ANSI 标准化后，C 语言的标准在相当长一段时间内都保持不变，尽管 C++不断在改进。C 标准在 20 世纪 90 年代才有了改进，这就是 ISO9899:1999（1999 年出版），这个版本就是通常所提及的 C99，它被 ANSI 于 2000 年 3 月采用。

目前，随着计算机技术的日益发展和微型计算机的普及，C 语言得以更为广泛地普及和应用。它广泛应用于系统软件、应用软件、数值计算和数据处理等各个领域，成为世界上应用极为广泛的程序设计语言。

1.1.2 C 语言的主要特点

C 语言具有传统程序设计语言的可靠性、简洁性和易使用性等优点，是一种结构化程序设计语言。其主要特点有：

(1) 程序结构简洁、紧凑、灵活。只需使用一些简单、规整的方法，就可以构造出复杂的数据类型或是功能很强的语句、程序。

(2) 表达能力强，有丰富的数据类型和运算符。C 语言不仅可以直接处理字符、数字、地址，还可以完成通常要由硬件才能实现的操作。另外，C 语言还允许用户自己定义数据类型。

(3) 生成的目标代码质量高。它将高级语言的基本结构和汇编语言的高效率结合起来，既具有高级语言易编程、易维护、可读性强、面向用户等特点，又具有汇编语言面向硬件的功能，可以编写系统软件。生成的目标代码的效率仅比汇编语言低 10～20%。

(4) 结构化的程序设计。C 语言具备编写结构化程序所需要的基本流程控制语句。程序设计的基本单元是函数，函数之间相互独立，从而实现了模块化的程序设计，提高了程序的可靠性。

(5) 良好的可移植性。用 C 语言编写的程序，其输入、输出功能通过调用函数实现，不依赖于硬件，因此，程序基本上不做修改就可用于不同型号的计算机和各种操作系统。

1.2 简单的 C 语言程序设计

下面以例 1.1 为例介绍 C 语言程序的基本结构和特点。

【例 1.1】 将两个从键盘上输入的整数相加，并把结果输出在计算机屏幕上。

```
#include <stdio.h>
int main(void)
{
    int x,y,sum;                    /* 定义三个整型变量 x,y,sum */
```

```
    scanf("%d %d",&x,&y);          /* 输入两个整数，并将其值赋给变量 x,y */
    sum=x+y;                       /* 计算两数之和*/
    printf("x+y=%d\n",sum);        /* 输出结果*/
    return 0;
}
```

该程序一共有 9 行，每行的作用如下所述。

1. #include 命令

程序的第 1 行#include <stdio.h>就是 include 命令，也叫做文件包含。一个程序可以有一条或多条 include 命令，也可以一条也没有。它的作用是告诉编译器，本程序中所用到的标准函数的函数原型是定义在哪个头文件中的。在编程时，对程序中用到的标准函数要使用#include 命令把相应的头文件包含到本文件中来，否则在编译时将出现所用函数未定义的错误。

stdio.h 是头文件的名字，大多数的 C 语言程序都要用到它，后缀.h 表明它是头文件(head)，当然不同的程序还可能使用其他的头文件。stdio.h 是标准输入输出头文件，使用它可以省去很多事，同时也能避免不必要的错误。因为它已经把要做的工作都“包含”进去了。

关于文件包含的具体使用，在以后的章节中将做详细的介绍。

2. main 函数

如第 2 行所示，对于一个 C 语言程序而言，无论大小，都是由一个或多个“函数”组成的。“函数”可以决定要完成的实际操作。但是，一个程序能运行出结果来，main 函数（也叫“主函数”）是必不可少的，并且一个程序只能有一个 main 函数。程序总是由 main 函数开始执行，并在 main 函数中结束，其他函数是由 main 函数直接或间接调用的。ANSI C 标准要求 main 函数的写法为 int main(void)，这将大大提高 C 程序的可移植性。一个最简单的 C 程序可以仅有主函数，如：

```
int main(void) { }
```

可见，花括号{ }也是 main 函数的组成部分。一个程序要完成的操作就放在里面，通常把花括号内的内容叫做函数体。上面主函数的函数体没有内容，因此它就什么也不做。在 main 后面有一对括号()，以后的程序中可能会看到在括号里面可以添加一些内容，但现在不用。尽管如此，括号也不能省略不写。函数详见第 5 章。

3. 变量定义

如第 4 行所示，程序中的 int x,y,sum;称为变量定义。和数学中的变量不同，C 程序中的变量必须在使用之前进行“定义”。定义的作用是：给不同的变量取不同的名字以示区别，为不同的变量分配不同的存储空间。

例 1.1 用 int x,y,sum;定义了三个变量，名字分别为：x，y，sum，变量的值是整数类型（用 int 来指定）。其中 x，y 分别存放加数和被加数，sum 存放两个数的和。

4. 输　入

C 语言有多种输入方式：可以从一个文件输入，也可以从键盘等设备输入。C 语言本身没有输入语句，通常采用系统提供的库函数 scanf 来完成输入。

程序中的第 5 行 scanf("%d %d",&x,&y);可以接收从键盘上任意输入两个整数。例如，输入：

13 7↙

此时，就将 13 赋值给变量 x，7 赋值给 y。输入时要注意两个数字之间必须要用空格分开。如果把语句改成：scanf("%d,%d",&x,&y);当再输入两个数时，数字之间则必须用逗号分隔。

除了 scanf 函数外，还有其他的库函数也能实现输入的功能。另外，不一定每个程序都要用到输入函数，编程时可以根据自己的需要来选择。

5. 程序语句

如第 6 行所示，程序是由一系列语句所组成的，这些语句描述了该程序要做的工作。每个语句都要完成一定的功能，每个语句都必须用分号;结束。比如，上面介绍的变量定义语句、输入语句等。

例 1.1 中的 sum=x+y;叫做赋值语句。它将 x 与 y 的值相加后赋值给变量 sum，这样就完成了求和的运算。

6. 输　出

如第 7 行所示，输入可以根据程序的需要用或是不用，但是一个程序如果没有输出信息，这个程序就不会有多大的使用价值。输出一般是指将一定形式的信息（文字或图形）写到屏幕、存储设备（软盘或硬盘）或输入/输出端口（串行口、打印机端口）等。

输出函数 printf 的功能是将输出信息显示在屏幕上，和前面的 scanf 一样，它也是由系统提供的标准函数之一。

如果 x，y 的值在输入时是 13 和 7，那么 printf("x+y=%d\n",sum);语句执行后，会在屏幕上显示：

x+y=20

7. main 函数的返回值

如第 8 行所示，表示 main 函数的返回值为 0。ANSI C 标准要求 main 函数必须返回一个 int 值给程序的激活者（通常是操作系统），其中 0 表示正常退出，非 0 表示出现异常。

8. 程序注释

如果在程序中对一些语句给出相应的解释，这可以提高程序的可读性和可维护性，这些解释称为注释。C 程序在进行编译的时候，注释部分会被忽略，因此，注释的内容可根据编程人员的需要任意书写，并不会带来任何使程序运行效率降低之类的问题。

C 程序中的注释内容用/*开头，用*/结束，在*和/之间不能有空格。程序在编译时，只要一遇到/*，就将以后的所有内容当做是注释，直到*/为止，也可用“//”注释从遇到“//”开始的一行。注释可以有一行，也可以有多行。在程序中加入适当的注释是一个良好的习惯，在大型程序中更有必要。

9. 编程风格

C 语言的书写非常自由，例 1.1 给出的程序也可以写成如下格式：

```
#include <stdio.h>
int main(void)
```

```
{ int x,y,sum; scanf("%d %d",&x,&y); sum=x+y;  printf("x+y=%d\n",sum); return 0;}
```

这种书写方式并不会影响程序的运行。也就是说，程序中的多条语句可以写在一行上，也可分成几行书写。显然，分行书写要清晰得多。

因此建议：每行写一条语句；主函数的一对花括号上下对齐；函数体内部采用缩进格式。当选定一种编程风格之后，一直用下去，从开始就养成一个良好的书写习惯。

1.3　算　法

1.3.1　算法的概念

一个程序通常包括数据结构和算法，数据结构是对数据的描述（如数据的类型及其组织形式），算法是对指定数据的操作方法和步骤，数据结构是程序的核心，而算法是程序的灵魂。

【例 1.2】 从键盘输入三个数，按从大到小的顺序输出这三个数。请给出解决这个问题的算法。

分析：程序对于从键盘输入的三个数必须用三个变量保存，设这三个变量分别为 a，b，c。先将 a 和 b 的值比较，若 a 小于 b，则将 a，b 值交换；再将 a 和 c 的值比较，若 a 小于 c，则将 a，c 值交换；最后将 b 和 c 的值比较，若 b 小于 c，则将 b，c 值交换。这样经过三次两两比较和交换，a 为最大值，c 为最小值，b 为中间一个数，最后将 a，b，c 顺序输出，即是按从大到小的顺序输出了这三个数（在将两个数进行交换时，还需要一个中间变量）。

算法步骤：

S1：输入三个数并将其值分别赋给三个变量 a，b，c；

S2：将 a 和 b 比较，若 a 小于 b，则将 a，b 值交换；

S3：将 a 和 c 比较，若 a 小于 c，则将 a，c 值交换；

S4：将 b 和 c 比较，若 b 小于 c，则将 b，c 值交换；

S5：输出 a，b，c 的值。

在上面的算法步骤中，第 2 步到第 4 步可详细描述，改进后的算法步骤为：

S1：输入三个数，其值分别赋给三个变量 a，b，c；

S2：将 a 和 b 比较，若 a<b，则 t=a，a=b, b=t；

S3：将 a 和 c 比较，若 a<c，则 t=a，a=c, c=t；

S4：将 b 和 c 比较，若 b<c，则 t=b，b=c, c=t；

S5：输出 a，b，c 的值。

这样，通过算法语言的描述，可以很方便地用程序语言来实现。注意：S1，S2…表示步骤 1、步骤 2……，S 为 Step 的简写。

1.3.2　算法的特性

算法一般具有以下特点：

（1）有穷性：任何算法必须在合理的时间内执行有限条指令后结束，也即一个算法的操

作步骤必须是有限的。

(2) 有效性：算法的每一个步骤都应是可执行的，正确的算法原则上都能精确地运行，并能得到正确的结果。

(3) 确定性：算法中的每一个步骤都必须是确定的，不能有歧义。

(4) 输入：算法一般都有一些输入的数据或初始条件，因此每个算法可能有零个或多个输入，如例 1.2 中输入的数据 a，b，c。

(5) 输出：每个算法都有一个或多个输出，如例 1.2 中输出的数据 a，b，c，没有输出的算法是无意义的。

1.3.3 算法的描述

算法的描述有多种方法。常用的描述方法有自然语言、流程图、伪代码等。

1. 自然语言

自然语言是指人们日常使用的语言，可以是汉语、英语或其他语言。用自然语言描述的算法通俗易懂，简单明了，如例 1.2，但如果算法中含有多种分支或循环操作时，自然语言就很难表述清楚，且冗长、容易产生歧义。

【例 1.3】 判断从 1900—2000 年中的每一年是否为闰年，并将结果输出。

闰年的条件：① 能被 4 整除，但不能被 100 整除的年份；② 既能被 100 整除，又能被 400 整除的年份。

假设 y 为年份，算法描述如下：

S1：将 1900 赋值给 y；

S2：若 y 不能被 4 整除，则输出 y“不是闰年”，然后转到 S6；

S3：若 y 能被 4 整除，不能被 100 整除，则输出 y“是闰年”，然后转到 S6；

S4：若 y 能被 100 整除，又能被 400 整除，则输出 y“是闰年”，然后转到 S6；

S5：输出 y“不是闰年”；

S6：将 y 的值加上 1 后重新赋值给 y；

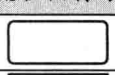
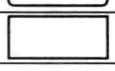
S7：当 y 的值小于或等于 2000 时，转到 S2 继续执行，否则算法结束。

这个算法中采用了循环与多次判断，与例 1.2 相比，根据这个算法编写程序难度会有所增加，因此，除了那些很简单的算法外，程序的算法一般不用自然语言描述。

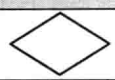
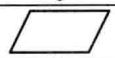
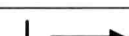
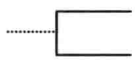
2. 流程图

流程图是用带箭头的线条将一些图框连接而成，用流程图来描述算法直观形象，易于理解。流程图的符号采用 ANSI 规定的一些常用的流程图符号，符号及其具体含义见表 1.1。

表 1.1 常用算法的流程图符号及其含义

流程图符号	名 称	含 义
	起止框	算法的开始或结束，每一个独立的算法只有一对起止框
	处理框	算法中的指令或指令序列，即对数据进行处理

续表 1.1

流程图符号	名　称	含　义
	判断框	对给定条件进行判断，若条件满足转向一个出口，否则转向另一出口
	输入输出框	算法中数据的输入或输出
	连接点	用于将画在不同地方的流程线连接起来，避免流程线的交叉或过长，如当一流程图在一页不能画完时，用它表示对应的连接处，用中间带数字的小圆圈表示，如①
	流程线	算法中流程的走向，连接上面的各种图形框
	注释框	不是流程图的必要部分，不反映流程和操作，只是对流程图中某些框的操作进行补充说明，帮助理解流程图

一般来说，流程图完全可用表 1.1 中的符号表示，流程线将各框图连接起来，它们的有序组合就构成了不同的算法描述。而对于结构化的程序，所有符号构成的流程图只包含 3 种基本结构：顺序结构、分支结构和循环结构，一个完整的算法可由这 3 种基本结构有机构成。

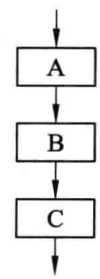


图 1.1　顺序结构的流程图

（1）顺序结构。

顺序结构是最简单的一种基本结构，根据流程线的方向按顺序执行各指定操作。其结构如图 1.1 所示，表示执行完 A 框中的操作后，必然紧接着执行 B 框中的操作，然后再执行 C 框中的操作，其执行顺序为从上到下，即 A→B→C。

（2）选择/分支结构。

选择/分支结构中必须包含一个判断框，根据其给定的条件 P 进行判断，由判断结果来确定执行 A 分支还是 B 分支，其中 A 或 B 中可以有一个为空。流程图的基本形状有两种，如图 1.2 所示。

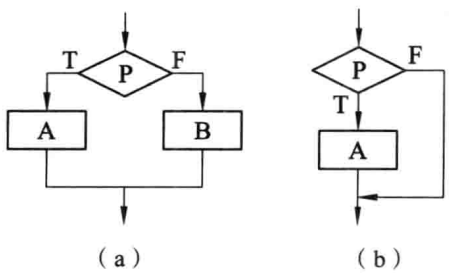


图 1.2　选择/分支结构流程图

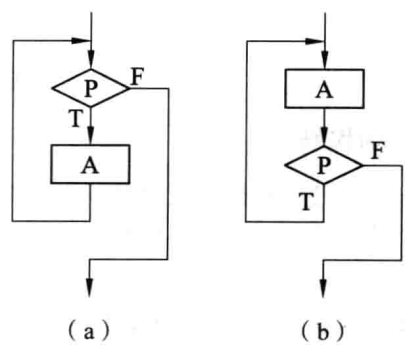


图 1.3　循环结构流程图

（3）循环结构。

循环结构是在条件为真的情况下，反复执行某一操作，其基本结构有两种：

① 当型循环结构。

如图 1.3（a）所示，其执行顺序为：先判断条件 P 是否成立，如果条件成立，执行 A，

然后再进入条件 P 的判断，若条件为真，继续执行 A，如此反复执行 A，当条件 P 不成立时，跳出循环。

② 直到型循环结构。

如图 1.3（b）所示，其执行顺序为：先执行 A，再判断条件 P 是否成立，若条件成立，则重复执行 A，然后再判断条件 P 是否成立，如此反复执行 A，直到给定的条件 P 不成立为止，即条件 P 一旦为假，则跳出循环。

【例 1.4】 将例 1.3 所描述的问题用流程图来表示，如图 1.4 所示。

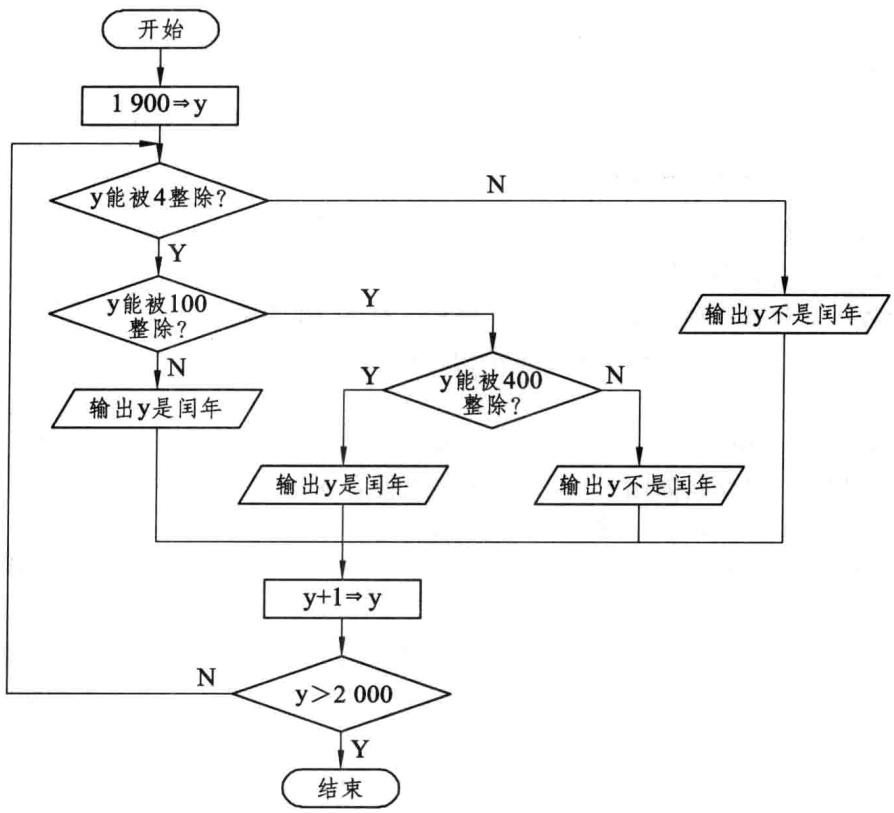


图 1.4 例 1.3 流程图

3. 伪代码

伪代码是一种接近程序语言的算法描述方法，用介于自然语言和计算机语言之间的文字和符号描述算法。伪代码既可用英文，也可以用中文，还可中英文混用，没有固定的语法规则，只要便于书写和阅读，且把意思表述清楚即可。

【例 1.5】 将例 1.3 的算法用伪代码来表示。

```
begin                                    /*算法开始*/
1 900⇒y
while (y≤2 000)
{
   if y 能被 4 整除
       if y 不能被 100 整除
```