21世纪高职高专规划教材

电子信息
工学结合模式
系列教材

C与C51程序设计项目教程

丁向荣 普清民 赖金志 编著
姚永平 主审

清华大学出版社

电子信息
工学结合模式
系列教材

21世纪高职高专规划教材

C与C51程序设计项目教程

丁向荣 普清民 赖金志 编著

清华大学出版社
北京

内 容 简 介

采用C语言编程是单片机应用、嵌入式系统应用编程必然的发展趋势。本书将C语言基本知识与Keil C有机结合在一起，既体现了电子信息大类专业方向的应用特色，又保留了C语言程序设计的通用性本色。本书采用任务驱动模式组织教材内容，将理论与实践紧密结合，易于实施“教、学、做”一体化教学模式，同时又便于读者自学与实践。

本教材可作为应用本科、高职高专、中职院校电子信息专业、电子通信专业、自动化专业、计算机相关专业C语言程序设计的教材，也可作为成人教育以及在职人员的培训教材、自学读物。

图书在版编目（CIP）数据

C与C51程序设计项目教程/丁向荣，普清民，赖金志编著. —北京：清华大学出版社，2014
21世纪高职高专规划教材. 电子信息工学结合模式系列教材
ISBN 978-7-302-34478-0

Ⅰ. ①C… Ⅱ. ①丁… ②普… ③赖… Ⅲ. ①C语言－程序设计－高等职业教育－教材
Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2013)第274294号

责任编辑：王剑乔
封面设计：傅瑞学
责任校对：刘　静
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795764
印 装 者：北京市密东印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：16　　**字　　数**：366千字
版　　次：2014年1月第1版　　**印　　次**：2014年1月第1次印刷
印　　数：1～2700
定　　价：32.00元

产品编号：054049-01

前　言

C语言是目前最为基础、最为流行的程序设计语言，具有简洁、紧凑、灵活、实用、高效、可移植性好等优点。C语言的数据类型丰富，可直接面向机器，既可用来编写系统程序，又可用来编写应用程序。单片机的C语言编程已成为单片机应用的必然趋势。本书新增了C51应用编程，着重介绍了C语言在8051单片机应用编程新增的数据类型、中断函数及开发工具，体现了C语言程序设计的具体应用，解决了传统计算机语言教学中“抽象，不知学有何用?”的弊病，能有效地提高学生的学习兴趣，为后续单片机、嵌入式系统的学习与应用奠定基础。

本教材根据工学结合的教学规律，采用以项目为导向、任务为驱动的教学模式组织教材内容，循序渐进。教材包括课程导引、C程序设计篇与C51应用篇3个部分。课程导引包括C语言的发展与主要特点、C程序的基本结构、程序的算法以及C语言集成开发环境的使用；C程序设计篇包括顺序程序设计、选择结构程序设计、循环结构程序设计、数组的应用、用函数实现模块化程序设计、指针的应用、构造用户自己的数据类型、编译预处理、文件9个项目；C51应用篇包括Keil C集成开发环境、C51应用编程两个项目。

C语言程序设计方面的教材有很多，相比其他教材，本教材具有如下特色：

(1) 新增C51应用篇，体现了C语言程序设计具体的应用特性，增加C语言程序设计课程与后续课程的连贯性。

(2) 采用以项目为导向、任务为驱动的教学模式组织教材内容，符合应用本科、高职高专、中职的教学目标，体现工学结合的职业教育教学特色。

本教材配有电子课件，以方便教学与读者自学使用。

本书由广东轻工职业技术学院丁向荣负责统筹、策划、统稿，具体编写课程导引、项目10与项目11；中山职业技术学院普清民编写项目3、项目5～8；广东轻工职业技术学院赖金志编写项目1、项目2、项目4与项目9。感谢深圳宏晶科技有限公司姚永平总经理对本教材的提议、建议与指导！在本书编写过程中参阅了大量书籍，同时也引用了互联网上的资料，在此向这些书籍和资料的原作者表示衷心的感谢！

限于编者水平有限，书中难免存在不当之处，恳请广大读者批评指正！任何批评、交流与建议，请发至：dingxiangrong65@163.com，不胜感谢！

编　者

2013年10月于广州

目 录

课程导引 …… 1

0.1 C语言的发展与主要特点 …… 1
0.1.1 计算机程序与计算机语言 …… 1
0.1.2 C语言的发展与主要特点 …… 2
0.2 C程序的基本结构 …… 4
0.3 程序的算法 …… 6
0.4 C语言集成开发环境的使用 …… 11
0.4.1 安装 Visual C++ 6.0 与运行 Visual C++ 6.0 集成开发环境 …… 12
0.4.2 单程序文件的操作步骤 …… 12
0.4.3 多程序文件的操作步骤 …… 15
思考与提高 …… 20

C程序设计篇

项目1 顺序程序设计 …… 23

任务 1.1 数据的表现形式及其运算 …… 23
任务 1.2 C语句的运用 …… 29
任务 1.3 数据的输入/输出 …… 31
思考与提高 …… 35

项目2 选择结构程序设计 …… 38

任务 2.1 if 语句实现的选择结构(一) …… 42
任务 2.2 if 语句实现的选择结构(二) …… 45
任务 2.3 用 switch/case 语句实现的多分支结构 …… 48
思考与提高 …… 51

项目3 循环结构程序设计 …… 54

任务 3.1 用 while 语句实现的循环结构 …… 54
任务 3.2 用 do-while 语句实现的循环结构 …… 56

任务 3.3 用 for 语句实现的循环结构 …… 58
任务 3.4 循环嵌套 …… 61
任务 3.5 goto、break、continue 语句的应用 …… 64
思考与提高 …… 67

项目 4 数组的应用 …… 70

任务 4.1 一维数组 …… 70
任务 4.2 二维数组 …… 74
任务 4.3 字符数组 …… 77
思考与提高 …… 80

项目 5 用函数实现模块化程序设计 …… 83

任务 5.1 函数的定义与调用 …… 83
5.1.1 函数的分类 …… 83
5.1.2 函数的定义 …… 84
5.1.3 函数的返回值 …… 86
5.1.4 函数的调用 …… 86
5.1.5 函数原型的声明 …… 87
任务 5.2 函数间的参数传递 …… 89
任务 5.3 函数间的嵌套与递归 …… 94
任务 5.4 变量的作用域和存储类别 …… 97
任务 5.5 内部函数和外部函数 …… 104
任务 5.6 库函数 …… 107
思考与提高 …… 109

项目 6 指针的应用 …… 113

任务 6.1 一维数组与指针 …… 113
任务 6.2 二维数组与指针 …… 119
任务 6.3 字符数组与指针 …… 121
思考与提高 …… 125

项目 7 构造用户自己的数据类型 …… 129

任务 7.1 结构体与结构体数组 …… 130
任务 7.2 结构体指针 …… 140
任务 7.3 共用体 …… 145
任务 7.4 枚举数据类型 …… 149
任务 7.5 用 typedef 定义类型 …… 152
思考与提高 …… 155

项目 8　编译预处理 …… 159

任务 8.1　宏定义 …… 159
任务 8.2　文件包含 …… 163
任务 8.3　条件编译 …… 166
思考与提高 …… 168

项目 9　文件 …… 170

任务 9.1　文件的基本操作 …… 170
任务 9.2　顺序读写数据文件 …… 173
任务 9.3　随机读写数据文件 …… 179
任务 9.4　文件读写的出错检测 …… 183
思考与提高 …… 185

C51 应用篇

项目 10　Keil C 集成开发环境 …… 189

任务 10.1　应用 Keil μVision4 开发工具编辑、编译用户程序生成机器代码 … 189
任务 10.2　应用 Keil μVision4 集成开发环境调试用户程序 …… 201
思考与提高 …… 209

项目 11　C51 应用编程 …… 211

任务 11.1　C51 基础 …… 211
任务 11.2　if、while、for、switch/case 语句的应用编程 …… 219
任务 11.3　C51 中断函数 …… 225
思考与提高 …… 228

附录一　ASCII 码表 …… 231

附录二　C 语言关键字 …… 232

附录三　C 语言的运算符种类、优先级与结合性 …… 233

附录四　常用头文件与库函数 …… 235

附录五　Keil C51 编译器扩展的关键字 …… 245

参考文献 …… 246

课 程 导 引

课程导引是C语言学习的重要的准备过程，一是了解C语言的作用、特点与发展历史；二是从宏观上掌握C语言源程序的组成结构；三是掌握C语言应用编程的开发过程与C语言开发工具，包括程序的编辑、编译、连接与运行。

主要内容：

(1) 计算机指令与程序的概念。

(2) 机器语言、汇编语言与高级语言。

(3) C语言的特点与C语言源程序的组成结构。

(4) 算法的概念与算法的描述。

(5) C语言源程序的处理过程。

重点与难点：

(1) C语言源程序的编辑、编译、连接与运行。

(2) 算法的概念与描述。

0.1 C语言的发展与主要特点

0.1.1 计算机程序与计算机语言

一个完整的计算机是由硬件和软件两部分组成的，缺一不可。看得到、摸得着的实体部分是计算机的硬件部分，计算机硬件只有在软件的指挥下，才能发挥其效能。计算机采取“存储程序”的工作方式，即事先把程序加载到计算机的存储器中，当启动运行后，计算机便自动地按照程序进行工作。

1. 指令与程序

计算机在人们眼中是“万能”的，能自动完成各种各样的工作。但究其本质，计算机只能完成一些简单的操作，计算机的每一次操作都是根据人们事先指定的指令进行的，通过简单操作的不同组合及快速运行，计算机就能按照人们的意志完成各种各样的工作。但计算机确实是最伟大的电子产品。

(1) 指令是规定计算机完成特定任务的命令，微处理器就是根据指令指挥与控制计算机各部分协调地工作。

(2) 程序是指令的集合，是解决某个具体任务的一组指令。在用计算机完成某个工作任务之前，人们必须事先将计算方法和步骤编制成由逐条指令组成的程序，并预先将它以二进制代码(机器代码)的形式存放在程序存储器中。

2. 编程语言

编程语言分为机器语言、汇编语言和高级语言。

(1) 机器语言是用二进制代码表示的，是机器能直接识别的语言，因此机器语言程序又称为目标程序。早期的计算机编程就是用二进制代码进行编程的，但机器语言与人们习惯的语言差别太大，难学、难写、难记忆、难阅读、难修改、难推广，当时只有极少数计算机专业人员会用机器语言编程。

(2) 汇编语言是用英文助记符来描述指令的，如用 ADD 表示“加”、SUB 表示“减”等，记忆、阅读、书写远胜于机器语言，但计算机并不能直接识别与执行汇编语言指令，需要用一种称为汇编程序的软件，将汇编语言指令转换为机器语言指令代码，才可被计算机识别与执行。助记符指令与机器代码指令有一一对应的关系，与机器语言指令一样，直接面向机器操作，依赖于具体机器的特性。机器语言与汇编语言都称为计算机的低级语言。

(3) 高级语言是一种接近于人们习惯使用的自然语言与数学语言的编程语言。20世纪50年代开发出了第一种计算机高级语言——Fortran 语言。数十年来，全世界涌现出了2500多种高级语言，每种高级语言都有其特定的用途，影响最大的有 Fortran 语言和 Algol(适合数值计算)、Basic/QBasic(适合初学者的小型会话语言)、Cobol(适合商业管理)、Prolog(人工智能语言)、C 语言(系统描述语言)、C++ 语言(支持面向对象程序设计的大型语言)、Visual Basic(支持面向对象程序设计的语言)等。

高级语言经历了以下几个不同的发展阶段。

(1) 非结构化语言。初期的高级语言都属于非结构化设计语言，编程风格比较随意，只要符合语法规则即可，程序中的流程可随意跳转，使程序变得难以阅读与维护。早期的 Basic、Fortran 等都属于非结构化设计语言。

(2) 结构化语言。规定程序必须由顺序结构、选择(分支)结构、循环结构等基本模块构成，程序中流程不允许随意跳转，程序总是由上而下顺序执行各个基本模块。这种程序具有结构清晰、易于编写、阅读和维护等特点。QBasic、Fortran 77 和 C 语言都属于结构化程序设计语言。

(3) 面向对象的语言。非结构化语言、结构化语言都属于基于工作过程语言，编写程序时需要具体指定每一个过程的细节，适用于编写较小规模的程序。在实践应用的发展中，人们又提出了面向对象的程序设计方法。程序面对的不是过程的细节，而是一个个对象，对象是由数据以及对数据进行的操作组成。C++、C#、Java 等语言是支持面向对象程序设计的语言。

0.1.2　C 语言的发展与主要特点

C 语言是目前国际广泛使用的高级语言。

1. C 语言的发展历程

C 语言的祖先是 BCPL 语言。BCPL 语言如何演化为 C 语言以及 C 语言的发展历程

如表 0-1 所示。

表 0-1 C 语言发展历程表

时 间	C 语言发展概况
1967 年	英国剑桥大学的 Martin Richards 提出了没有类型的 BCPL(Basic Combined Programming Language)语言
1970 年	美国 AT&T 贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,设计出了很简单且很接近硬件的 B 语言。但 B 语言过于简单且功能有限
1972—1973 年	美国 AT&T 贝尔实验室的 D. M. Ritchie 在 B 语言基础上设计出了 C 语言。C 语言既保持了 BCPL 和 B 语言的优点(精练、接近硬件),又克服了它们的缺点(过于简单、无数据类型)。开发 C 语言的目的是尽可能地降低编程对硬件平台的依赖性,使之具有移植性。C 语言的新特点主要体现在具有多种数据类型(如字符、数值、数组、指针等)
1973 年	最初的 C 语言是为描述和实现 UNIX 操作系统提供一种工作语言,Ken Thompson 和 D. M. Ritchie 合作把 UNIX 的 90%以上用 C 语言改写。随着 UNIX 的日益广泛使用,C 语言迅速得到推广
1978 年	Brain W. Kernighan 和 Dennis M. Ritchie 合著了影响深远的名著《The C Programming Language》,这本书介绍的 C 语言实际成为第一个 C 语言标准。1978 年以后,C 语言先后移植到大、中、小和微型计算机上,C 语言很快风靡全世界,成为世界上应用最广泛的程序设计语言
1983 年	美国国家标准协会(ANSI)根据 C 语言问世以来各种版本对 C 语言的发展和扩充,制定了第一个 C 语言标准草案('83 ANSI C)
1989 年	美国国家标准协会公布了一个完整的 C 语言标准——ANSI X3. 159-1989(常称为 ANSI C 或 C89)
1990 年	国际标准化组织 ISO 接收 C89 作为国际标准 ISO/IEC 9899:1990(简称 C90),它和 ANSI 的 C89 基本上是相同的
1999 年	1995 年,ISO 对 C90 做了一些修订,1999 年又对 C 语言标准进行修订,在基本保留原来 C 语言特性的基础上,针对应用的需要,增加了一些功能,尤其是 C++ 中的一些功能,命名为 ISO/IEC 9899:1999,2001 年与 2004 年先后进行了两次技术修正。ISO/IEC 9899:1999 及其技术修正被称为 C99 标准

2. C 语言的特点

C 语言既可以编写系统软件,又可以编写应用软件,其主要具有以下特点。

(1) 语言简洁、紧凑,使用方便、灵活。只有 37 个关键字、9 种控制语句,程序书写形式自由,一行中可书写多条语句,一个语句可分散在多行。

说明:虽然 C 语言书写形式自由,为了便于阅读、维护,建议在学习与应用编程中,养成良好的书写习惯。

(2) 运算符丰富。C 语言有 34 种运算符,把括号、赋值、强制类型转换等都作为运算符处理,表达式类型多样化。

(3) 数据类型丰富。其包括整型、浮点型、字符型、数组类型、指针类型、结构体类型、共用体类型等,C99 又扩充了复数浮点类型、超长整型(long long)、布尔类型(bool)。

(4) 模块化结构。具有结构化的控制语句,如 if/else 语句、while 语句、do/while 语句、

switch/case语句、for语句等，用函数作为程序的基本模块单位，便于实现程序的模块化。

(5) 语法限制不太严格，程序设计自由度大。如对数组下标越界不做检查，对变量的类型使用比较灵活。因此，不能完全依赖编译查错，程序员更要养成严谨的工作习惯，仔细检查，确保自己的程序正确。

(6) 允许直接访问物理地址，能进行位操作，可以直接对硬件进行操作。C语言具有高级语言的功能和低级语言的许多功能，这种双重性使它既是成功的系统描述语言，又是通用的程序设计语言。

(7) 用C语言编写的程序可移植性好。C语言的编译系统简洁，很容易移植到新系统，在新系统上运行时，可直接编译“标准链接库”中的大部分功能，不需要修改源代码。几乎所有计算机系统都可以使用C语言。

(8) 生成目标代码质量高，程序执行效率高。

C语言既可以编写系统软件，又可以编写应用软件。许多以前只能用汇编语言处理的问题，现在都可以改为用C语言来编程了，如各种单片机、嵌入式系统应用编程都采用C语言编程了，C51篇所介绍的就是专门针对8051单片机的C语言编程知识。

0.2 C程序的基本结构

1. C语言程序结构形式

```
#include<stdio.h>                          //包含命令
#define PI  3.1415                         //宏定义
int time;                                  //全局变量定义
float fun_1(int a, int b);                 //函数声明
/*---自定义函数1----*/
float fun_1(int a, int b)                  //函数首部
{                                          //函数体
    声明部分
    执行部分
}
      ⋮
/*---自定义函数n----*/
int fun_2(int x,int y)                     //函数首部
{                                          //函数体
    声明部分
    执行部分
}
/*---主函数----*/
void main(void)                            //函数首部
{                                          //函数体
    声明部分
    执行部分
}
```

2. C程序结构说明

一个C程序包括3大部分，即预编译命令、全局声明和函数定义。

(1) 预编译命令包括文件包含命令(#include)、宏定义与宏定义的撤销(#define、#undef)和条件编译(#if、#else、#endif)。

(2) 全局声明包括变量声明和函数声明,全部变量声明是指在函数之外进行变量声明,即在函数外定义的变量为全局变量,反之在函数内部定义的变量称为局部变量;当一个函数调用另一个函数时,被调用函数必须事先声明,被调用函数的声明既可以在调用函数中声明,也可以在调用函数的前面进行声明,在函数外部声明时,一般放在预编译命令之后、函数定义之前处声明。

(3) 函数是C语言程序的基本单位,一个C语言程序可包含多个不同功能的函数,但一个C语言程序中只能有一个且必须有一个名为main()的主函数。主函数的位置可在其他功能函数的前面、之间或最后。当功能函数位于主函数的后面位置时,在主函数调用时必须"先声明"。

C语言程序总是从main()主函数开始执行。主函数可通过直接书写语句或调用功能子函数来完成任务。功能子函数可以是C语言本身提供的库函数,也可以是用户自己编写的函数。

(4) 注释。注释不是C程序所必需的,只是为了便于阅读而设置。有两种注释方式。

① 以//开始的单行注释,这种注释可以单独占一行,也可以出现在一行中其他内容的右侧。

② 以/*开始、以*/结束的块式注释,这种注释可以包含多行内容。编译系统会将一个/*开始符与下一个*/结束符之间的内容作为注释。

3. 函数结构

一个函数包括两部分,即函数首部与函数体。

(1) 函数首部。函数首部即为函数的第一行,包括函数类型、函数名、函数参数类型、函数参数名。

(2) 函数体。函数体是指函数首部下方花括号内的部分,又分为声明部分和执行部分。

① 声明部分包括定义在本函数中所用到的变量和对本函数所调用函数的声明。

② 执行部分由若干个语句组成,指定在函数中所进行的操作。

4. 库函数与自定义函数

库函数是针对一些经常使用的算法,经前人开发、归纳、整理形成的通用功能子函数。ANSI C提供了100多个标准库函数,不同的C编译系统除提供标准库函数外,还提供一些专门的应用函数,如Keil C则包含了针对8051单片机应用编程的库函数。

自定义函数是用户自己根据需要而编写的子函数。

【例0-1】 下面的程序是"输入三角形3条边,求面积"的C语言源程序(EX0-1.C),试分析其程序结构。

```
1   #include<stdio.h>
2   #include<math.h>
3   float fun_area(int x,int y,int z)      //定义求"已知三角形3条边求面积"的子函数
```

```
4   {
5       float s,temp;
6       s=(x+y+z)/2;
7       temp=sqrt(s*(s-x)*(s-y)*(s-z));
8       return(temp);
9   }
10  void main(void)
11  {
12      int a,b,c;
13      float area;
14      scanf("%d,%d,%d",&a,&b,&c);          //从键盘输入三角形的3条边
15      area=fun_area(a,b ,c);               //调用"已知三角形3条边求面积"的子函数
16      printf("area=%f\n",area);            //输出三角形的面积
17  }
```

解：

(1) 程序首部有两条包含语句，包含了 stdio.h 和 math.h 两个头文件。因为主函数调用的 scanf()、printf()(输入/输出)函数在 stdio.h 头文件中；自定义函数 fun_area()调用的 sqrt()(求平方)函数在 math.h 头文件中。

(2) 包含一个主函数 main()和一个子函数 fun_area()，主函数调用了 fun_area()子函数，fun_area()子函数位于主函数之前定义，符合“先定义、后使用”的函数调用原则。

0.3　程序的算法

一个程序主要包括以下两方面的信息。

(1) 对数据的描述。在程序中要指定用到哪些数据以及这些数据的数据类型和组织形式，这也就是数据结构。

(2) 对操作的描述。在程序中指定计算机操作的步骤，也就是算法。

数据是操作对象，操作的目的是对数据进行加工处理。作为程序设计人员，必须认真考虑和设计数据结构和操作步骤(即算法)。著名计算机科学家沃思(Nikiklaus Wirth)提出一个公式，即

算法＋数据结构＝程序

实际上，一个过程化的程序除了以上两个主要因素外，还应当采用结构化程序设计方法来进行程序设计，并且用一种计算机语言来描述。因此，算法、数据结构、程序设计方法和计算机语言等4个方面是一个程序设计人员所应具备的知识。

算法是解决“做什么”和“怎么做”的问题。程序中的操作语句，实际上就是算法的体现。

1. 算法的概念

广义地说，为解决一个问题而采取的方法和步骤称为算法。

计算机程序的算法可分为两大类别：数值运算算法和非数值运算算法。数值运算的目的是求数值解，如求平方根、求圆柱体的体积等，都属于数值运算范畴。非数值运算包

括的面非常广，最常见的是用于事务管理领域，如学生档案管理、职工工资管理、图书管理等。

数值运算往往有现成的模型，可采用数值分析的方法，因此对数值运算算法的研究比较深入，算法比较成熟。对各种数值运算都有比较成熟的算法可供选用，如计算机程序系统中的“数学程序库”，C 语言编译系统中的头文件 math.h 中就包含了许多数学运算算法。

非数值运算的种类繁多，要求各异，难以做到全部都有现成的答案，只有一些典型的非数值运算算法（如排序、查找搜索算法）有现成、成熟的算法可供选用。大多问题需要程序设计者参照已有的类似算法思路，自行进行设计相关问题算法。

【例 0-2】 求 1+2+3+4+5，试编制求解算法。

解：

① 用最原始的方法实现。

步骤 1：先求 1+2，得到结果 3。

步骤 2：将步骤 1 得到的结果 3 与 3 相加，得到结果 6。

步骤 3：将步骤 2 得到的结果 6 与 4 相加，得到结果 10。

步骤 4：将步骤 3 得到的结果 10 与 5 相加，得到结果 15。

试想用这种方法求解 1+2+…+100，需要编写多少个步骤？99 个步骤，显然是不可取的。

② 寻找一种通用的运算算法。

设置两个变量，i 为被加数，j 为加数。此外，每次运算的和直接存回被加数变量。用循环算法来求解结果。可将上述算法修改如下：

步骤 1：i=1。

步骤 2：j=2。

步骤 3：i+j→i。

步骤 4：j+1→j。

步骤 5：若 j≤5，则返回重新执行步骤 3、步骤 4 与步骤 5；否则，算法结束。

当采用这种算法后，求解 1+2+…+100 时，只需将步骤中的 5 改为 100 即可。同样，当遇到类似规律的计算时，只需做些小改动即可。如计算 1×2×3×4×5，只需将算法中步骤 3 中的“+”号改为“×”号即可。因此，这种算法具有很强的通用性与灵活性。

注意： 对计算机而言，第②种算法并没有减少它的运行步数，反而是增加了，但当运行数据增加时，大大地减少编写程序步数，将第①种算法中的顺序运行改为了循环运行。计算机是高速数据处理机器，循环运行是轻而易举的，因此，循环运行是程序算法的重要形式，循环程序结构是程序设计最为重要的结构。

2. 算法的描述

为了描述一个算法，可以有多种方法，主要有自然语言、传统流程图、N-S 流程图。

1）用自然语言描述算法

自然语言就是人们日常使用的语言，如汉语、英语或其他语言。用自然语言描述通俗易懂，但文字冗长，容易出现歧义。一般情况下，不建议使用自然语言描述算法。

2）用传统流程图描述算法

传统流程图，简称流程图。流程图是用一些图形框来描述各种操作，直观形象，易于理解。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，如图0-1所示。

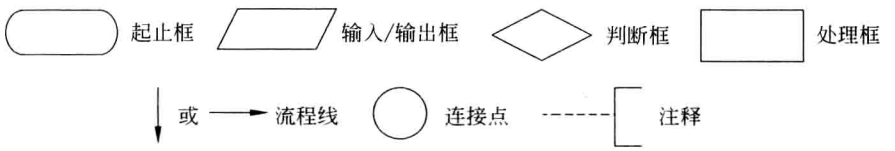

图0-1 流程图各种图框与名称

结构化程序设计有3种基本程序结构，有关它的流程图描述如下。

（1）顺序结构。如图0-2所示，虚线框内是一个顺序结构，其中A和B两个框是顺序执行的。即，执行A后，必定执行B。

（2）选择结构。选择结构又称为分支结构，如图0-3所示。此结构中必包含一个判断框。根据给定的条件p是否成立选择执行A框或B框。A框和B框是相互独立的，无论条件p是否成立。只能执行A框或B框之一。A框或B框中可以有一个是空的，即不执行任何操作，如图0-4所示中B框是空的。

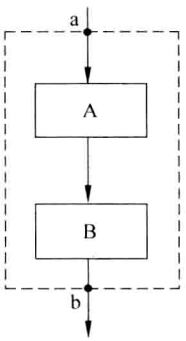

图0-2 顺序结构流程图

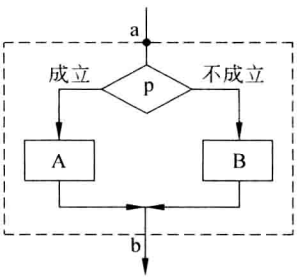

图0-3 选择结构流程图(1)

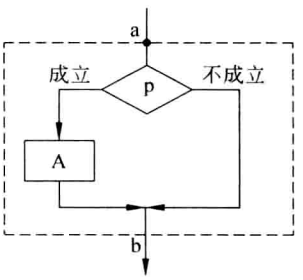

图0-4 选择结构流程图(2)

（3）循环结构。循环结构又称重复结构，可分为两类结构。

① 当型(while型)循环结构。当型循环结构如图0-5(a)所示。当给定的条件p_1成立时，执行A框操作，执行完A后，再判断条件p_1是否成立，若成立继续执行A框操作，如此反复执行A框操作，直至某次条件p_1不成立为止，从而从b点退出本循环结构。

② 直到型(until型)循环结构。直到型循环结构如图0-5(b)所示。先执行A框操作，然后判断给定的条件p_2是否成立，若条件p_2不成立，再执行A，然后再判断条件p_2是否成立，若条件p_2仍不成立，再执行A，如此反复执行，直至给定条件p_2成立为止，从而从b点退出循环结构。

3）用N-S流程图描述算法

1973年，美国学者I. Nassi和B. Shneiderman提出了一种新的流程图形式，完全去掉

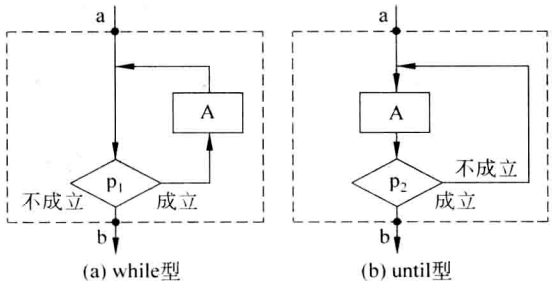

图 0-5　循环结构流程图

了带箭头的流程线，全部算法写在一个矩形框内，在这个框内包含从属于它的框，这种流程图称为 N-S 结构化流程图，简称 N-S 流程图。

用 N-S 流程图描述结构化程序的 3 种基本结构。

(1) 顺序结构。如图 0-6 所示，由 A 框和 B 框组成。

(2) 选择结构。选择结构如图 0-7 所示，当 p 条件成立时执行 A 操作，当 p 条件不成立时执行 B 操作。

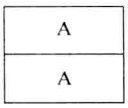

图 0-6　顺序结构的 N-S 流程图

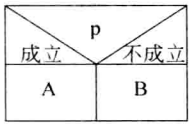

图 0-7　选择结构的 N-S 流程图

(3) 循环结构。当型循环结构如图 0-8 所示，当 p_1 条件成立时反复执行 A 操作，直至 p_1 条件不成立为止；直到型循环结构如图 0-9 所示，先执行 A 的操作，再判断 p_2 条件，只要 p_2 条件不成立再反复执行 A 操作，直到 p_2 条件成立为止。

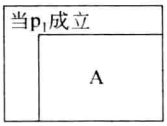

图 0-8　当型循环结构的 N-S 流程图

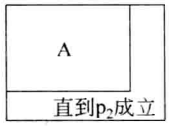

图 0-9　直到型循环结构的 N-S 流程图

【例 0-3】 判断 2000—2050 年中的每一年是否为闰年，并将结果输出。试用自然语言、传统流程图及 N-S 流程图描述其算法。

解：首先要分析闰年的条件：

能被 4 整除，但不能被 100 整除的是闰年，有 1996 年、2008 年、2012 年、…、2048 年；

能被 400 整除的是闰年，有 2000 年；

不符合这两个条件的不是闰年。

根据闰年的条件要求以及题目要求，设计的算法如下：

① 用自然语言描述。设 year 为年份变量。

步骤 1：2000→year。

步骤2：若year不能被4整除，则输出year和“不是闰年”字符，然后转到步骤6，检查下一个年份。

步骤3：若year能被4整除，不能被100整除，则输出year和“是闰年”字符，然后转到步骤6，检查下一个年份。

步骤4：若year能被400整除，则输出year和“是闰年”字符，然后转到步骤6，检查下一个年份。

步骤5：输出year和“不是闰年”字符。

步骤6：year＋1→year。

步骤7：当year≤2500时转步骤2继续执行，否则算法停止。

② 用传统流程图描述。如图0-10所示。

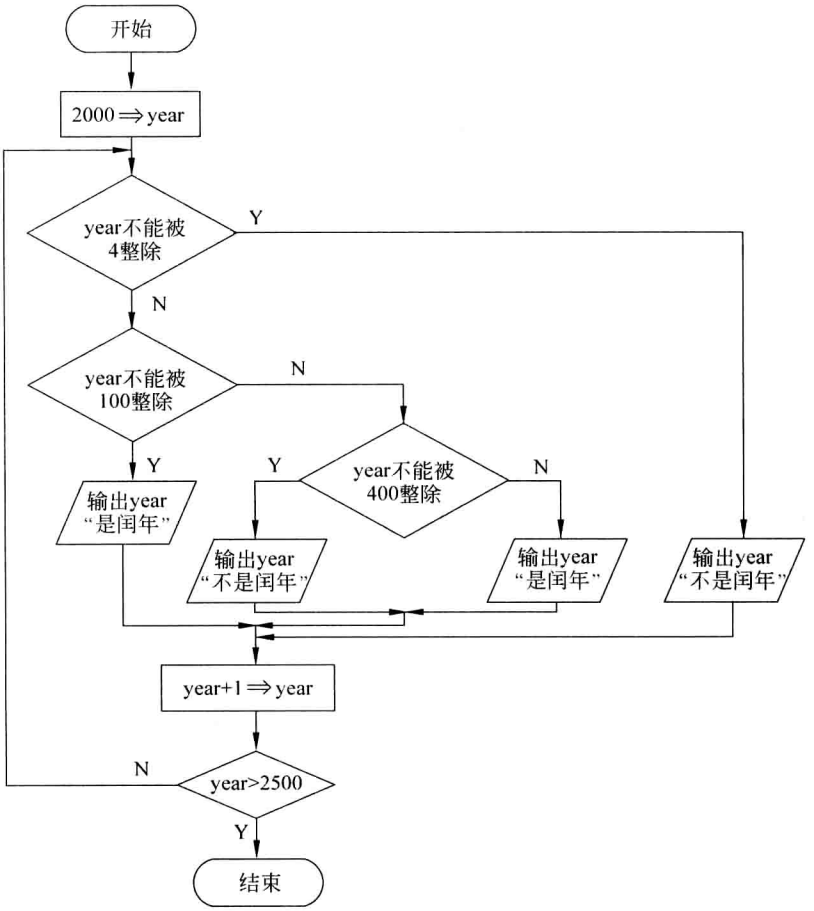

图0-10　例0-3算法的传统流程图

③ 用N-S流程图描述。如图0-11所示。