

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

C 语言程序设计 实例教程

朱家义 黄勇 主编

朱艳丽 李莉 炎士涛 副主编

清华大学出版社



21 世纪普通高校计算机公共课程规划教材

C 语言程序设计 实例教程

朱家义 黄勇 主编

朱艳丽 李莉 炎士涛 副主编

清华大学出版社
北京

内 容 简 介

本书以程序设计知识板块为基础,针对高级语言的知识体系结构,围绕学习目标为相关内容设计出相应实例任务,通过与任务相关知识的学习,达到完成该章任务的目的。各章均安排具有针对性的拓展实例,以便于读者进一步理解和牢固掌握各章节的基本内容,体现了学以致用原则。

全书共分 11 章,主要内容包括算法基础、C 语言基础、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组与指针、函数、文件、编译预处理、简单图形编程和综合应用程序的开发等内容。

本书可作为高等学校计算机及相关专业程序设计的教材,也可作为计算机等级考试以及计算机爱好者自学用书和各类工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计实例教程/朱家义,黄勇主编. —北京:清华大学出版社,2010.1
(21 世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-19382-1

I. C… II. ①朱… ②黄… 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(2009)第 012471 号

责任编辑:梁 颖 林都嘉

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:19.75 字 数:481 千字

版 次:2010 年 1 月第 1 版 印 次:2010 年 1 月第 1 次印刷

印 数:1~3000

定 价:29.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:030860-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评

审确定主题。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

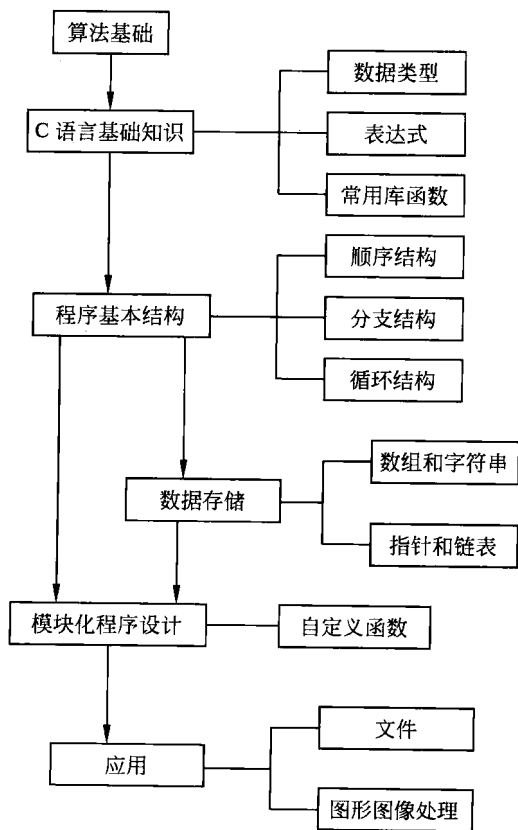
联系人：梁颖 liangying@tup.tsinghua.edu.cn

前言

程序设计是计算机科学及相关专业的基础课程，是学习后续课程的基础和工具。本教材体现了三基（基础理论、基本知识、基本技能）、三性（科学性、启发性、适用性）和少说（只介绍基本概念、原理和方法）多练（加强语法和算法思想的培养）原则，使之符合不同层次的人才培养目标和培养模式。

本教材具有以下特色：

（1）以高级语言的知识体系结构为基础，把握全书知识板块（如下图所示），学以致用，为后续课程如“数据结构”、“编译原理”、“操作系统”、“图形图像处理”等打下基础。



（2）每章均有导读和小结，便于学生抓住要领。每章的导读内容中，简要地介绍本章内容及知识点。

（3）体现了任务驱动式教学思想。围绕学习目标设计出相应实例任务，通过问题分析

引导出该章内容，通过对与任务相关的知识学习，达到完成该章任务的目的。

(4) 精选实例，进行思维拓展。各章均安排具有针对性的拓展实例，以便于读者通过练习，进一步理解和掌握各章节的基本内容。

(5) 基于结构化程序设计思想进行编程。程序设计上，完全按照结构化程序设计思想进行编程，进行规范化程序设计训练。

(6) 安排了内容丰富的习题。通过练习加深对所学内容的理解，进一步掌握相关算法的特点和适用范围，提高程序设计能力。

(7) 强化训练。教材的最后一章安排了一个综合程序设计，全面应用了所学内容。

本书由朱家义、黄勇任主编，朱艳丽、李莉和炎士涛任副主编。朱家义编写第1章，炎士涛编写第2章，黄勇编写第10章和附录，李莉编写第6章，张伟编写第7章，朱艳丽编写第3、4章，郭晓娟编写第5章，马丽娟编写第8、9章，梁云娟编写第11章，全书最后由朱家义统稿。

在本书的编写过程中，得到有关领导和清华大学出版社的大力支持，在此一并表示感谢！

由于作者水平所限，书中难免有不当之处，敬请各位同仁和读者批评指正。联系方式：liangying@tup.tsinghua.edu.cn。

作 者

2009年8月

目 录

第 1 章 程序设计基础	1
1.1 算法	1
1.1.1 算法的定义	1
1.1.2 算法的描述方法	1
1.1.3 算法设计的要求	6
1.1.4 算法时间效率	6
1.1.5 结构化程序设计	8
1.2 问题求解的实现过程	9
1.3 C 语言程序结构	10
1.3.1 C 程序的结构	10
1.3.2 标识符和关键字	13
本章小结	14
习题 1	15
第 2 章 C 语言基础知识	17
2.1 数据类型简介	17
2.2 常量和变量	19
2.2.1 常量	19
2.2.2 符号常量	19
2.2.3 变量	20
2.3 基本数据类型	20
2.3.1 整型	20
2.3.2 实型	21
2.3.3 字符型	22
2.3.4 枚举类型	24
2.4 构造类型	26
2.4.1 数组	26
2.4.2 结构体	27
2.4.3 共用体	30
2.4.4 用 typedef 定义类型	32
2.5 指针	33

2.5.1	指针与指针变量	33
2.5.2	指针变量的定义	34
2.5.3	指针变量的引用	35
2.6	运算符与表达式	36
2.6.1	算术运算	36
2.6.2	关系和逻辑运算	37
2.6.3	赋值运算	39
2.6.4	位运算	41
2.6.5	条件运算	42
2.6.6	逗号运算	42
2.6.7	表达式的其他说明	43
2.7	数据类型转换	43
2.7.1	自动类型转换	44
2.7.2	赋值类型转换	44
2.7.3	强制类型转换	44
2.8	常用库函数介绍	45
2.8.1	数学函数	45
2.8.2	字符函数	47
	本章小结	48
	习题 2	48
第 3 章	顺序结构程序设计	51
3.1	任务描述	51
3.2	赋值语句	52
3.3	数据输入与输出	53
3.3.1	printf 函数	53
3.3.2	scanf 函数	60
3.3.3	字符数据的输入与输出	64
3.4	程序实现	66
3.5	实例拓展	67
	本章小结	68
	习题 3	69
第 4 章	选择结构程序设计	72
4.1	任务描述	72
4.2	if 语句	73
4.2.1	单分支 if 语句	73
4.2.2	双分支 if 语句	74
4.2.3	if 语句的嵌套	75

4.3	switch 语句	76
4.4	程序实现	78
4.5	实例拓展	79
	本章小结	82
	习题 4	82
第 5 章	循环结构程序设计	87
5.1	任务描述	87
5.2	while 循环	88
5.3	do-while 循环	90
5.4	for 循环	91
5.5	几种循环的比较	95
5.6	break 语句、continue 语句和 goto 语句	96
5.6.1	break 语句	96
5.6.2	continue 语句	97
5.6.3	goto 语句	99
5.7	循环嵌套	100
5.8	实例拓展	102
	本章小结	104
	习题 5	105
第 6 章	数组与指针	109
6.1	任务描述	109
6.2	一维数组	110
6.2.1	一维数组的初始化	111
6.2.2	一维数组的应用	112
6.3	二维数组	114
6.3.1	二维数组的初始化	115
6.3.2	二维数组的应用	116
6.4	字符数组与字符串	116
6.4.1	字符数组的初始化和字符串的操作	116
6.4.2	字符串的应用	121
6.5	数组与指针	122
6.5.1	指针与一维数组	123
6.5.2	指针与二维数组	126
6.5.3	指针与字符串	127
6.5.4	指针数组	129
6.6	链表	130
6.6.1	动态存储分配	130

6.6.2 链表的概念	132
6.7 程序实现	136
6.8 实例拓展	137
本章小结	140
习题 6	140
第 7 章 函数与指针	144
7.1 任务描述	144
7.2 函数概述	145
7.2.1 模块化程序设计方法	145
7.2.2 函数的定义	146
7.2.3 函数的调用	147
7.2.4 函数参数的进一步说明	148
7.2.5 函数返回值的进一步说明	150
7.2.6 被调用函数的声明和函数原型	151
7.3 函数间的参数传递	152
7.3.1 值传递	152
7.3.2 地址传递	152
7.3.3 数组作为函数参数	153
7.4 函数的嵌套调用与递归调用	160
7.4.1 函数的嵌套调用	160
7.4.2 函数的递归调用	161
7.5 变量的作用域和存储方式	164
7.5.1 变量的作用域	164
7.5.2 变量的存储类别	167
7.5.3 内部函数和外部函数	170
7.6 指针与函数	171
7.6.1 指向函数的指针	171
7.6.2 返回指针的函数	173
7.6.3 带参数的主函数	175
7.7 程序实现	176
7.8 实例拓展	177
本章小结	181
习题 7	181
第 8 章 文件	185
8.1 任务描述	185
8.2 文件概述	186
8.2.1 文件的概念	186

8.2.2 文件的分类	186
8.3 文件操作	187
8.3.1 FILE 文件类型指针	187
8.3.2 文件的打开操作	188
8.3.3 文件的关闭操作	190
8.3.4 文件的读写操作	191
8.4 文件的定位	199
8.4.1 置文件位置指针于文件开头的函数 <code>rewind</code>	199
8.4.2 改变文件位置指针的函数 <code>fseek</code>	200
8.4.3 取得文件当前位置的函数 <code>ftell</code>	201
8.4.4 文件的错误检测	202
8.5 程序实现	203
8.6 实例拓展	204
本章小结	208
习题 8	208
第 9 章 预处理命令	211
9.1 任务描述	211
9.2 宏定义	212
9.3 文件包含	219
9.4 条件编译	221
9.5 程序实现	224
9.6 实例拓展	224
本章小结	225
习题 9	226
第 10 章 简单图形编程	229
10.1 任务描述	229
10.2 基本函数	229
10.3 常用绘图函数	232
10.3.1 画点函数	232
10.3.2 画线函数	232
10.3.3 设定线型函数	233
10.3.4 任意封闭图形的填充	235
10.3.5 有关图形窗口和图形屏幕操作函数	236
10.4 实例拓展	240
本章小结	243

第 11 章 C 语言综合程序设计	244
11.1 应用程序设计的一般过程	244
11.2 概要设计	245
11.2.1 学生信息管理模块	245
11.2.2 学生成绩管理模块	246
11.2.3 系统帮助功能	246
11.3 详细设计及编码	246
11.3.1 声明和数据设计	246
11.3.2 主函数设计	248
11.3.3 学生信息管理模块	251
11.3.4 学生成绩管理模块	265
11.4 程序测试	278
本章小结	279
附录 A ASCII 码表	280
附录 B Turbo C 集成开发环境	281
附录 C Turbo C 2.0 常用库函数	291
附录 D Turbo C 语言运算符表	301
参考文献	302

第 1 章

程序设计基础

利用高级语言编写程序，必须掌握算法的设计和表示，然后根据实际问题的求解过程，按照所使用的高级语言的语法规则，表示成程序。C 语言具有强大的功能，能够灵活方便地解决各种实际问题。

本章主要内容：

- 算法的定义；
- 算法的表示方法；
- 算法的三种基本结构；
- 问题求解的实现过程；
- C 语言程序结构。

1.1 算 法

一个程序应包括以下两方面内容：(1)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。(2)对操作的描述。即操作步骤，也就是算法。

1.1.1 算法的定义

算法是求解问题的步骤，它是指令的有限序列，其中每一条指令表示一个或多个操作。

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。数值运算的目的是求数值解，例如求方程的根，求一个函数的定积分等，都属于数值运算范围。非数值运算包括的面十分广泛，最常见的是用于事务管理领域，例如图书检索、人事管理、行车调度管理等。

算法具有下列 5 个特性：

(1) 有穷性。一个算法必须对任何合法的输入在执行有穷步之后结束，且每一步都可在有限时间内完成。

(2) 确定性。算法中每一条指令必须有确切的含义，不会产生二义性理解，并在任何条件下，算法只有唯一的一条执行路径。

(3) 可行性。算法中描述的操作都可以通过已经实现的基本运算执行有限次来实现。

(4) 输入。一个算法有零个或多个的输入，这些输入取自于某个特定的对象的集合。

(5) 输出。一个算法有一个或多个的输出。

1.1.2 算法的描述方法

要描述一个算法，必须清晰地写出每一步应该干什么，下面以“从 10 个数中求出最

大数”为例介绍几种常用的描述方法。

1. 用伪代码描述

所谓伪代码,指的是一种介于自然语言和计算机语言之间的一种符号,它可以是英文单词或英文单词的缩写。用这种代码描述算法时,简单方便、清晰易懂,便于向程序转换。

算法如下:

① input x	/*输入一个数,并把该数存入 x 中*/
② max<=x	/*把 x 的值送入 max 中*/
③ n=0	/*设置一个计数器 n,并置初值为 0*/
④ if n=9 goto ⑨	/*如果 n 的值等于 9,则转入第⑨步执行*/
⑤ input x	/*输入一个数,并把该数存入 x 中*/
⑥ if max<x then max<=x	/*如果 max 的值小于 x 的值,则把 x 的值送入 max 中*/
⑦ n=n+1	/*计数器 n 增加 1*/
⑧ goto ④	/*转向第④步执行*/
⑨ output max	/*输出 max 的值*/

由于本题是从 10 个数中选取最大数,就可以采取比较的方法进行筛选。为此,在第①步和第②步中首先输入了一个数,并把这个数存入到一个用来保存最大数的变量中。

从第④步到第⑧步构成了一个重复执行的部分,为了控制重复的次数,并能在有限次内结束,在第③步特别设置了一个计数器 n,并赋初值为 0。

由于在第①步已输入了一个数据,所以再输入 9 个数即可。为此,只需让重复执行的部分执行 9 次,所以当 n=9 时,就应该跳出重复执行的部分,转向第⑨步。

第⑤步和第⑥步使得每输入的一个新数据和 max 比较,并让 max 总是保存已输入数据中的最大数。

第⑦步完成的是一个计数工作,每输入一个数,就让 n 增加 1。然后由第⑧步转向第④步执行,从而形成了重复执行。

当重复执行结束后,由第④步转向第⑨步执行,并由第⑨步输出 max 中的值(最大值),至此整个算法结束。

2. 用流程图描述

流程图是用一些图框和流程线来表示各种操作及其操作顺序。用这种方法表示算法,直观形象、易于理解。流程图中常用的基本图形如图 1-1 所示。



图 1-1 流程图中常用的基本图形

1966 年, Bohra 和 Jacopini 提出了算法的三种基本结构,并用这三种基本结构作为描述算法的基本单元。

1) 顺序结构

顺序结构表示的是算法按照操作步骤描述的顺序依次执行的一种结构,用流程图来描

述如图 1-2 所示, 按照操作步骤描述的顺序, 依次执行 A, B, C, …各部分。算法语言中没有专门实现顺序结构的控制语句。

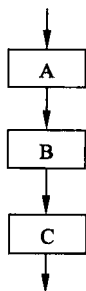


图 1-2 顺序结构

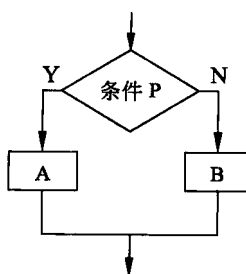


图 1-3 分支结构

2) 分支结构

分支结构表示的是按照条件的成立与否决定程序执行不同的方式。本节例子的伪代码描述的算法中第⑥步就是分支结构。图 1-3 所示为分支结构的流程图。

在分支结构中, 根据条件 P 的判断结果, 决定执行 A 或执行 B, 不能既执行 A 又执行 B。无论执行 A 或执行 B, 都要经过一个出口脱离分支结构。在许多情况下, B 框允许是空的, 即不执行任何操作。

算法语言中通常设置专门的控制语句来实现分支结构。如 C 语言中的 if 语句。

3) 循环结构

循环结构又称重复结构, 它实现重复执行某一部分的操作。本节例子的伪代码描述的算法中第④步到第⑧步之间的部分就是重复执行的部分。重复执行的部分也称为循环体。按循环判断条件和循环体出现的先后次序, 可分为两类循环方式。

(1) 当型循环。先进行条件判断, 然后确定循环体是否执行, 如图 1-4 (a) 所示。

它的执行过程如下:

- ① 判断条件 P 是否成立, 若不成立转向④执行; 若成立, 执行②。
- ② 执行循环体 A。
- ③ 转向①执行。
- ④ 退出循环。

根据它的执行过程可知, 当循环条件 P 在第一次判断时不成立, 则循环体一次也不执行。

(2) 直到型循环。先执行一次循环体, 然后根据条件是否成立来决定循环体是否继续执行, 如图 1-4 (b) 所示。

它的执行过程如下:

- ① 执行循环体 A。
- ② 判断条件 P 是否成立, 若不成立转向③执行; 若成立, 执行①。
- ③ 退出循环。

根据它的执行过程可知, 循环体 A 至少要执行一次。

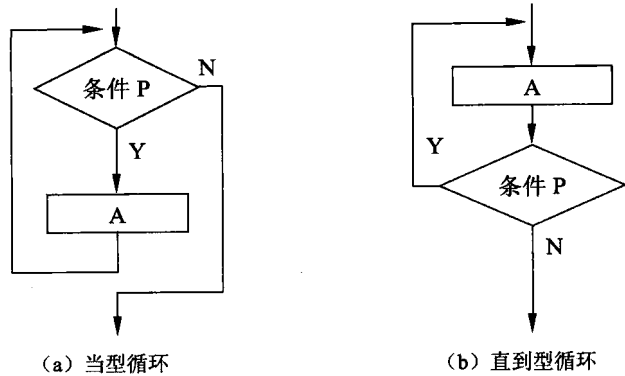


图 1-4 循环结构

本节例子的算法流程图如图 1-5 所示。

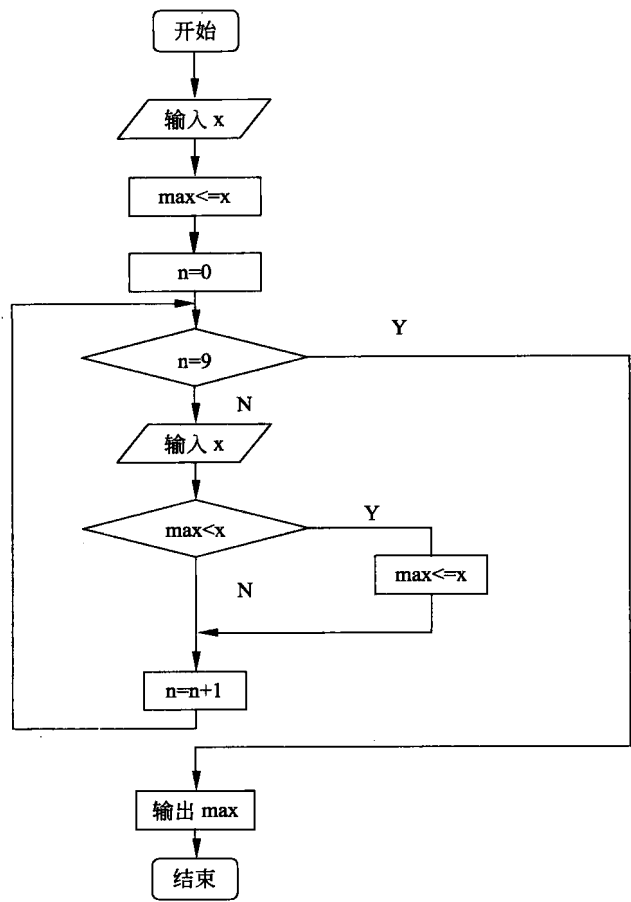


图 1-5 用流程图描述的算法

3. 用 N-S 图描述

流程图由一些特定意义的图形、流程线及简要的文字说明构成，它能清晰明确地表示算法中各步骤之间的关系和执行顺序。在使用过程中，人们发现流程线不一定是必需的，