

中等职业教育计算机系列教材

C 语言程序设计实用教程

曾令明 编著

电子科技大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计实用教程/曾令明编著. —成都: 电子科技大学出版社, 2006.5
ISBN 7-81114-128-0

.C... 曾... .C 语言 - 程序设计 - 专业学校 - 教材 .TP312

中国版本图书馆 CIP 数据核字 (2006) 第 044758 号

内 容 提 要

全书系统地讲解了程序设计和算法的基本概念、C 语言程序设计基础知识、基本结构程序设计、函数与预处理命令、数组的应用、结构体与共用体、指针的应用、文件定义及相关函数等知识, 并安排了课程设计供学生练习。本书在内容安排上根据学生的实际需要, 力求浅显易懂, 并以丰富的实例来帮助学生理解所学知识。

本书为中等职业教育教材, 也可作为各类计算机培训班的培训教材, 以及各类职业院校学生的教材或参考书。

C 语言程序设计实用教程

曾令明 编著

出 版: 电子科技大学出版社 (成都建设北路二段四号 邮编: 610054)

责任编辑: 罗 雅

发 行: 新华书店经销

印 刷: 成都墨池教育印刷总厂

开 本: 787mm × 1092mm 1/16 印张 15.75 字数 383 千字

版 次: 2006 年 5 月第一版

印 次: 2006 年 5 月第一次印刷

书 号: ISBN 7 - 81114 - 128 - 0/TP · 30

定 价: 22.00 元

本书如有印刷、装订等质量问题, 请寄回印刷厂调换
版权所有不得翻印

前 言

社会的发展离不开高素质的专业人才，随着我国现代化建设的深入进行，职业教育正逐步体现其在国民经济建设中的重要作用。按照《中共中央国务院关于深化教育改革全面推进素质教育的决定》精神，以及《面向 21 世纪教育振兴行动计划》中提出的职业教育课程改革和教材建设规划的要求，当前的职业教育要“以职业能力为本位”，“以就业为导向”，服务于培养高素质劳动者和技能人才这一目标，要围绕培养技术应用能力这条主线来设计学生的知识、能力、素质结构，加强对学生的基本实践能力、专业技术应用能力及综合实践能力的培养。在职业教育使用的教材中适当增加实验或案例所占的比重，在教学中多采用“项目教学法”和“案例教学法”，引导学生积极思考，主动参与，勤于实践。

为了使理论与实践更好地结合，培养学生分析问题和解决问题的能力，从整体上提升中等职业学校学生的计算机应用水平，促进教学改革，我们依据教育部制定的《中等职业学校计算机及应用专业教学指导方案》，以及教育部等六部委最新制定的《中等职业学校计算机应用与软件技术专业领域技能型紧缺人才培养培训指导方案》，组织一批职教专家和优秀教师，结合最新的教学改革研究成果，编写了这套中等职业教育计算机系列教材。

本书为本套中等职业教育计算机系列教材之一，每章的内容都是以最基础的知识为起点，让学生熟练掌握书中的基本概念和基本操作，然后进入更深层次的学习，最后根据所附的习题和上机实训，进行学习检测和操作练习。

全书系统地讲解了程序设计和算法的基本概念、C 语言程序设计基础知识、基本结构程序设计、函数与预处理命令、数组的应用、结构体与共用体、指针的应用、文件定义及相关函数等知识，并安排了课程设计供学生练习。本书在内容安排上根据学生的实际需要，力求浅显易懂，并以丰富的实例来帮助学生理解所学知识。

本书编者是长期从事计算机教学的一线教师，具有丰富的教学实践经验。本书凝聚了编者多年来的教学经验和成果，编写上深入浅出，通俗易懂，注重所讲知识的实用性和可操作性，着重培养学生的动手能力。

本书为中等职业教育教材，也可作为各类计算机培训班的培训教材，以及各类职业院校学生的教材或参考书。本教材相关学习资料及习题答案可登录 www.21pcedu.com 免费下载。

由于编者水平有限，书中难免存在不妥之处，敬请广大读者批评指正。

编 者
2006 年 5 月

目 录

第 1 章 概述.....	1
1.1 程序设计的基本概念.....	1
1.1.1 程序.....	1
1.1.2 程序设计.....	1
1.1.3 程序设计语言.....	2
1.2 算法.....	3
1.2.1 算法概念.....	3
1.2.2 算法的描述.....	5
1.3 C 语言简介.....	9
1.3.1 C 语言的产生及发展.....	9
1.3.2 C 语言的特点.....	9
1.3.3 C 程序组成及书写格式.....	10
1.3.4 C 程序上机过程.....	11
习题.....	13
上机实验.....	13
第 2 章 C 语言程序设计基础.....	15
2.1 关键字与标识符.....	15
2.1.1 关键字.....	15
2.1.2 标识符.....	16
2.2 数据类型.....	16
2.2.1 C 的数据类型.....	16
2.2.2 C 的基本数据类型.....	17
2.3 常量.....	18
2.3.1 整型常量.....	18
2.3.2 实型常量.....	19
2.3.3 字符型常量.....	20
2.3.4 符号常量.....	22
2.4 变量.....	23
2.4.1 变量的定义.....	23
2.4.2 变量的基本操作.....	24
2.4.3 整型变量.....	24
2.4.4 实型变量.....	25

2.4.5	字符变量	25
2.5	运算符与表达式	26
2.5.1	运算符概述	27
2.5.2	算术运算符与算术表达式	27
2.5.3	赋值运算符与赋值表达式	28
2.5.4	表达式中的类型转换	29
2.5.5	逗号运算符与逗号表达式	31
	习题	32
	上机实验	34
第 3 章	基本结构程序设计	35
3.1	顺序结构	35
3.1.1	语句	35
3.1.2	数据的输入输出	37
3.2	选择结构	44
3.2.1	关系运算与逻辑运算	44
3.2.2	if 语句	47
3.2.3	switch 语句	50
3.3	循环结构	52
3.3.1	while 语句	52
3.3.2	do-while 语句	53
3.3.3	for 语句	55
3.3.4	循环的嵌套	58
3.3.5	break 语句与 continue 语句	59
	习题	61
	上机实验	67
第 4 章	函数与预处理命令	68
4.1	模块化程序设计与函数	68
4.1.1	模块化程序设计的概念	68
4.1.2	函数分类	69
4.2	函数的定义与调用	70
4.2.1	函数定义	70
4.2.2	函数调用	71
4.3	函数的嵌套与递归	78
4.3.1	函数的嵌套调用	78
4.3.2	函数的递归	79
4.4	变量的作用域	81

4.4.1	局部变量	81
4.4.2	全局变量	83
4.5	静态存储变量与动态存储变量	85
4.5.1	静态存储变量	85
4.5.2	动态存储变量	86
4.6	预处理命令	89
4.6.1	宏定义	89
4.6.2	文件包含	96
	习题	97
	上机实验	100
第 5 章	数组	102
5.1	数组概述	102
5.1.1	数组的引入	102
5.1.2	数组的概念	102
5.2	一维数组	103
5.2.1	一维数组的定义	103
5.2.2	一维数组的引用	104
5.2.3	一维数组的初始化	105
5.2.4	一维数组应用举例	106
5.3	二维数组	107
5.3.1	二维数组的定义	107
5.3.2	二维数组的引用	108
5.3.3	二维数组的初始化	109
5.3.4	二维数组应用举例	110
5.4	字符数组	111
5.4.1	字符数组的定义	111
5.4.2	字符数组的初始化	112
5.4.3	字符串	112
5.4.4	字符数组的引用	113
5.4.5	字符数组的输入和输出	113
5.4.6	字符串处理函数	115
5.4.7	字符数组应用举例	119
5.5	数组作为函数参数	119
5.5.1	数组元素作函数实参	119
5.5.2	数组名作函数参数	120
	习题	122

上机实验.....	126
第 6 章 结构体与共用体.....	127
6.1 结构体概述.....	127
6.1.1 结构体的引入.....	127
6.1.2 结构体类型的定义.....	127
6.2 结构体变量.....	129
6.2.1 结构体变量的定义.....	129
6.2.2 结构体变量的引用.....	131
6.2.3 结构体变量的初始化.....	133
6.2.4 结构体变量作函数参数.....	134
6.3 结构体数组.....	135
6.3.1 结构体数组的定义.....	135
6.3.2 结构体数组的初始化.....	135
6.3.3 结构体数组元素的引用.....	136
6.3.4 结构体数组应用举例.....	136
6.4 共用体.....	137
6.4.1 共用体类型的定义.....	138
6.4.2 共用体变量的定义.....	138
6.4.3 共用体变量的引用.....	139
6.5 位运算与位段.....	140
6.5.1 位运算.....	140
6.5.2 位段.....	141
6.6 枚举类型.....	143
6.7 用 typedef 定义类型.....	145
习题.....	146
上机实验.....	149
第 7 章 指针.....	150
7.1 地址和指针的概念.....	150
7.2 指针变量的定义与使用.....	151
7.2.1 指针变量的定义.....	151
7.2.2 指针变量的使用.....	151
7.3 指针与函数.....	154
7.3.1 指针作函数的参数.....	154
7.3.2 返回值为指针的函数.....	156
7.3.3 指向函数的指针.....	157
7.4 指针与数组.....	158

7.4.1 指向一维数组的指针	159
7.4.2 指向多维数组的指针	166
7.4.3 指针与字符串	170
7.4.4 指针数组和指向指针的指针	174
7.5 指针与结构体	179
7.5.1 指向结构体变量的指针	179
7.5.2 链表	181
7.5.3 指向结构体数组的指针	187
7.5.4 用指向结构体的指针作函数参数	188
习题	189
上机实验	194
第 8 章 文件	195
8.1 文件概述	195
8.1.1 文件的定义	195
8.1.2 文件的分类	195
8.2 文件指针	197
8.3 文件的打开与关闭	198
8.3.1 文件打开函数 (fopen)	198
8.3.2 文件关闭函数 (fclose)	199
8.4 文件的读写	200
8.4.1 字符读写函数 fgetc 和 fputc	200
8.4.2 字符串读写函数 fgets 和 fputs	203
8.4.3 数据块读写函数 fread 和 fwrite	205
8.4.4 格式化读写函数 fscanf 和 fprintf	207
8.5 文件的定位	209
8.5.1 fseek 函数	209
8.5.2 rewind 函数	210
8.5.3 ftell 函数	211
8.6 文件检测函数	212
8.6.1 feof 函数	212
8.6.2 ferror 函数	212
8.6.3 clearerr 函数	212
习题	212
上机实验	214
第 9 章 课程设计	215
9.1 课程设计的步骤	215

9.2 学生成绩管理系统设计	215
9.2.1 设计目的	215
9.2.2 设计要求	216
9.2.3 算法分析	218
9.2.4 参考源程序	220
9.2.5 重要知识点与功能扩展	239
9.2.6 评价标准	240
附录	241

第 1 章 概 述

学习目标

1. 了解程序设计的基本概念。
2. 了解程序设计语言的分类。
3. 掌握算法、算法的设计和算法的描述方法。
4. 熟悉 C 语言程序组成。

随着科学技术的迅猛发展，计算机技术日新月异，计算机程序设计语言也层出不穷。那么，什么是程序设计语言？应该学哪一种程序设计语言？如何进行程序设计？这些都是程序设计初学者首先遇到的问题。本章就程序设计的基本知识作概括性讨论，重点介绍算法的概念、特征、设计算法的方法和策略、流程图的表示和结构化程序设计方法等内容，同时还介绍了 C 语言的总体特征。需要说明的是，有些概念和方法要随着后续各章的深入学习才会有深刻的理解，不必在一时不懂的问题上停滞不前。

1.1 程序设计的基本概念

1.1.1 程序

从计算机诞生起，就有了程序。计算机通过执行程序完成其工作，如计算、控制、文字处理、图形处理、网络通信等。所谓程序，就是按照具体要求产生的适合计算机处理的指令序列。我国颁布的《计算机软件保护条例》对程序的概念给出了更为精确的描述：“程序是指为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列，或者可被自动地转化成代码化指令序列的符号化序列，或者符号化语句序列。”这就是说，程序要有目的性和可执行性。

1.1.2 程序设计

程序设计是指完成程序的过程，通常包含以下几个步骤：

- (1) 提出问题，明确程序要达到的要求。
- (2) 确定数据结构和所采用的算法。
- (3) 编制程序。
- (4) 调试程序。
- (5) 整理并写出文档。

1.1.3 程序设计语言

语言是用于传达信息的表示方法、约定和规则的集合,是人们交流信息的工具和媒介。而程序设计语言是实现程序设计,以便于人与计算机进行信息交流的工具和媒介。

程序设计语言一般分为低级语言和高级语言两大类,其中低级语言包括机器语言和汇编语言。

1. 机器语言

机器语言是以二进制形式表示的机器指令的集合,是计算机系统不需要翻译就可以直接识别和执行的程序设计语言。用机器语言编写程序,程序人员必须熟悉机器指令的二进制代码,记忆指令代码能完成的操作,还应指出这一操作对象的位置,即记忆指令的操作码和地址码。由于机器语言程序是直接计算机上执行,所以效率比较高,能充分发挥计算机的高速计算的能力。在计算机发展的初期,人们都是使用机器语言直接编制程序。但机器语言不易记忆和理解且缺乏直观性,所以用机器语言编制程序的难度很大。

2. 汇编语言

汇编语言用助记符来表示指令的操作码和操作对象,用标号和符号来代替地址、常量和变量。例如,“ADD AX,BX;”代表两个寄存器数相加的功能。用汇编语言编写的程序,较直观、易理解。但计算机却不能识别和直接运行汇编语言程序,必须由汇编程序将汇编语言程序“翻译”成机器语言程序后才能识别并运行。此外,不同 CPU 的指令系统对应的汇编语言不同。如:单板机、单片机、微处理器等,随机器型号、类型的不同,各自的汇编语言也不同。这就是说,用汇编语言编写的程序缺乏通用性。

3. 高级语言

为了方便使用,程序设计语言朝着接近人们熟悉、习惯的自然语言和数学语言描述的高级化方向发展,形成了各种各样、丰富多彩的程序设计高级语言。

高级语言是一类面向问题的程序设计语言,且独立于计算机的硬件,其表达方式接近于被描述的问题,易于人们的理解和掌握。用高级语言编写程序,可简化程序编制和测试,其通用性和可移植性好。目前,计算机高级语言虽然很多,据统计已经有好几百种,但广泛应用的却仅有十几种,他们有各自的特点和使用范围。如: BASIC 语言,是普及性会话语言中的一类; FORTRAN 语言,多用于科学及工程计算; COBOL 语言,多用于商业事务处理和金融业; PASCAL 语言,它有利于结构化程序设计; C 语言,常用于软件的开发; PROLOG 语言,多用于人工智能;而当前流行的,还有面向对象的程序设计语言 C++和 Java 等。

用高级语言编写的程序称为源程序,源程序不能被计算机直接执行,必须将他们“翻译”成具体机器的机器语言程序才能执行。高级语言“翻译”成机器语言有两种方式:一种是“解释”,一种是“编译”。C 语言属于编译型语言。编译的原理就是由编译程序把源程序编译、连接成可执行文件,然后由机器直接执行。C 语言源程序的运行过程如图 1-1 所示。

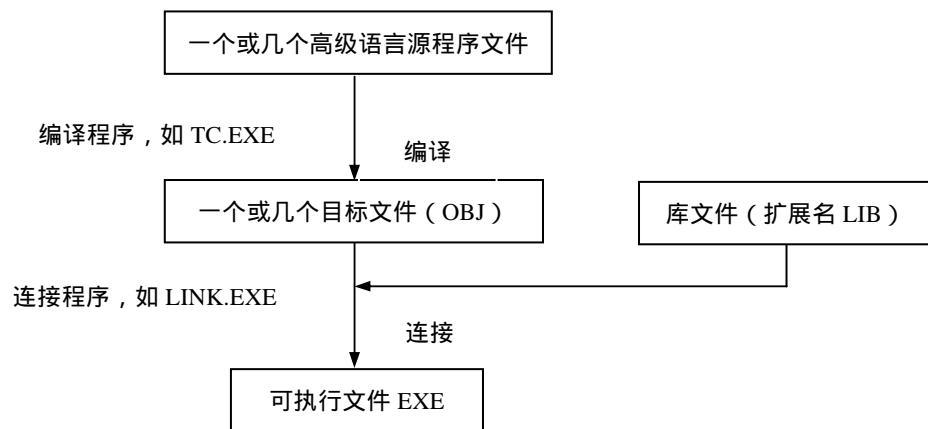


图 1-1 C 语言源程序的运行过程

1.2 算 法

1.2.1 算法概念

所谓算法，就是指为解决某一特定的问题，所给出的一系列确切的、有限的操作步骤。程序实际上是用程序设计语言所描述的算法。也就是说，依据算法所给定的步骤，用程序设计语言所规定的表达形式去实现这些步骤，即得到源程序。由此可见，算法是程序设计的关键之一。

下面通过例子来介绍如何设计一个算法：

【例 1-1】找出 A、B、C 三个数中的最大数，并把它放在变量 MAX 中。

由于计算机一次只能比较两个数，先把 A 与 B 比较，较大的数放入 MAX 中，再把 MAX 与 C 比，又把较大的数放入 MAX 中。

最后，把 MAX 输出，此时 MAX 中装的就是 A、B、C 三数中最大的数。算法可以表示如下：

- (1) 输入 A、B、C。
- (2) A 与 B 中大的一个放入 MAX 中。
- (3) 把 C 与 MAX 中大的一个放入 MAX 中。
- (4) 输出 MAX。

其中的(2)、(3)两步仍不明确，无法直接转化为程序语句，可以继续细化：

把 A 与 B 中大的一个放入 MAX 中。若 $A > B$ ，则 $MAX = A$ ；否则 $MAX = B$ 。

把 C 与 MAX 中大的一个放入 MAX 中。若 $C > MAX$ ，则 $MAX = C$ 。

于是算法最后可以写成：

- (1) 输入 A, B, C。

(2) 若 $A > B$, 则 $MAX = A$; 否则 $MAX = B$ 。

(3) 若 $C > MAX$, 则 $MAX = C$ 。

(4) 输出 MAX 。

这样的算法就可以很方便地转化为相应的程序语句。

【例 1-2】猴子吃桃问题：有一堆桃子不知数目，猴子第一天吃掉一半，觉得不过瘾，又多吃了一只，第二天照此办理，吃掉剩下桃子的一半另加一个，天天如此，到第十天早上，猴子发现只剩一只桃子了，问这堆桃子原来有多少个？

此题粗看起来有些无从着手的感觉，那么怎样开始呢？假设第一天开始时有 a_1 只桃子，第二天有 a_2 只，...，第 9 天有 a_9 只，第 10 天是 a_{10} 只。在 a_1, a_2, \dots, a_{10} 中，只有 $a_{10}=1$ 是知道的，现要求 a_1 。可以看出， a_1, a_2, \dots, a_{10} 之间存在一个简单的关系：

$$a_9 = 2*(a_{10}+1)$$

$$a_8 = 2*(a_9+1)$$

...

$$a_1 = 2*(a_2+1)$$

也就是：

$$a_i = 2*(a_{i+1}+1) \quad i = 9, 8, \dots, 1$$

这就是此题的数学模型。

观察上面从 a_9, a_8 直至 a_1 的计算过程，这其实是一个递推过程。这种递推的方法在计算机解题中经常用到。另一方面，这九步运算从形式上完全一样，不同的只是 a_i 的下标而已。由此，可以引入循环的处理方法，并统一用 a_0 表示前一天的桃子数， a_1 表示后一天的桃子数，算法可以写成如下：

(1) $a_1 = 1$; { 第 10 天的桃子数， a_1 的初值 }

$i = 9$ 。 { 计数器初值为 9 }

(2) $a_0 = 2*(a_1+1)$ 。 { 计算当天的桃子数 }

(3) $a_1 = a_0$ 。 { 将当天的桃子数作为下一次计算的初值 }

(4) $i = i - 1$ 。

(5) 若 $i >= 1$, 转 (2)。

(6) 输出 a_0 的值。

根据上面的例子，可以看出，在算法设计中应遵循下面几个准则：

(1) 输入。算法必须有零个或多个输入，以产生初始值。上面例 1-1 是从外界得到数据，例 1-2 是在算法中直接设定了需要的数据，可以看成预先输入了数据。

(2) 输出。算法必须有确定的结果，可产生一个或多个输出。

(3) 可行性。算法中的每一操作步骤必须能够用程序设计语言来描述，并能够被计算机实现。如除数为 0 的除法操作是无意义的，不满足可行性准则。

(4) 确定性。算法中的每一步操作必须有确切的含义，不会产生歧义。也就是说，该操作对于相同的输入只能得到相同的结果。

(5) 有穷性。算法必须在执行有限步骤以后结束，且每一步在有限时间内完成，算法不能出现“死循环”。

1.2.2 算法的描述

算法的描述方法有自然语言、伪代码、流程图、N-S图、PAD图等。这里只介绍容易掌握、容易理解的流程图和N-S图。

1. 流程图

流程图是一种传统的算法描述方法。它简单直观，应用广泛，特别是在早期语言阶段，只有通过流程图才能简明地表述算法，它成为程序员们交流的重要手段，直到结构化的程序设计语言出现，对流程图的依赖才有所降低。

流程图采用不同的几何图形来描述算法的逻辑结构，每个几何图形表示不同性质的操作。图1-2所示的是ANSI规定的常用流程图符号。

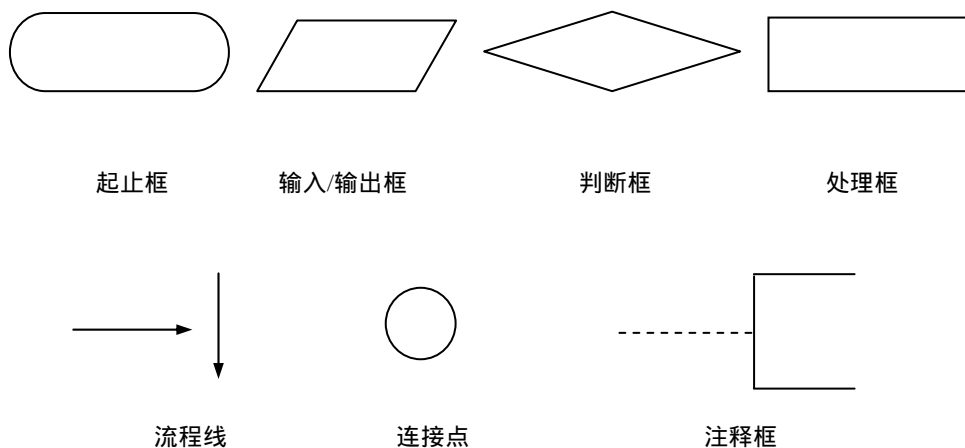


图 1-2 常用流程图符号

具体使用规定如下：

- (1) 起止框表示算法的起始、终止。
- (2) 输入/输出框表示要输入数据和输出结果。
- (3) 判断框内填判断条件，在判断框的左、右、下侧的流程线旁标注“真”、“假”或“T”、“F”或“是”、“否”或“Y”、“N”来表示判断条件成立与不成立时的运行方向。
- (4) 处理框用于计算或赋值等处理操作。
- (5) 流程线表示算法中操作的执行顺序，向下向右可不画箭头，其他方向的箭头必须画出。
- (6) 圈内标识同一字母的连接点表示流程图中的同一点，通常用于不同页之间流程图的连接或较远距离的流程线连接。
- (7) 注释框用文字说明注释信息，帮助理解流程图。

图1-3是例1-1中算法的流程图描述，图1-4是例1-2中算法的流程图描述。

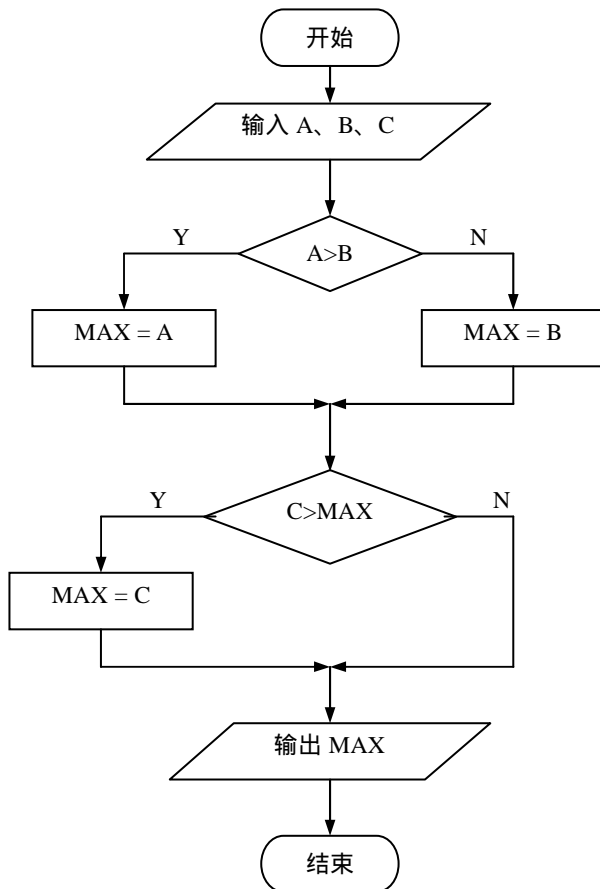


图 1-3 例 1-1 算法流程图

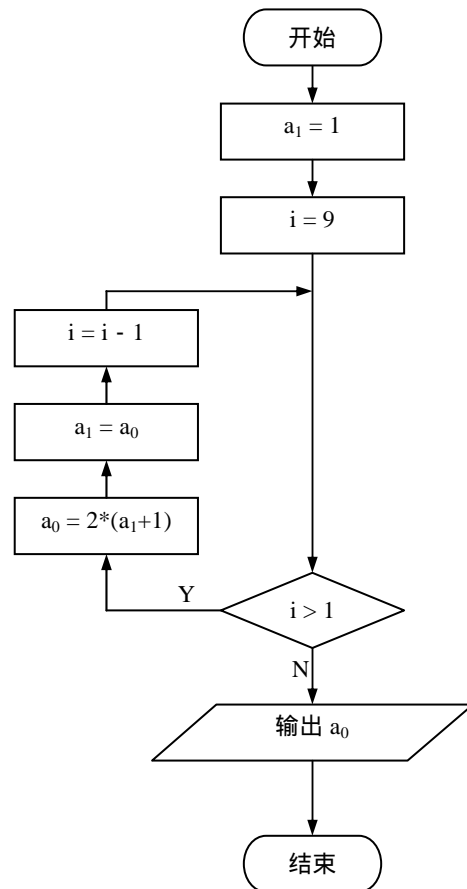


图 1-4 例 1-2 算法流程

2. 结构化程序设计

早期的高级语言中都有 goto 语句，它允许程序从一个地方直接跳转到另一个地方。这样做的优点是程序设计方便灵活，降低了人工复杂度，但其缺点也十分突出，一大堆跳转语句使得程序的流程十分复杂紊乱，难以看懂也难以验证程序的正确性，如果有错，排起错来更是十分困难。后来有人证明，任何复杂的算法都可以由顺序结构、选择（分支）结构和循环结构这三种基本结构组成。于是结构化程序设计应运而生。

仅用顺序、选择和循环三种基本结构来实现算法、编写程序的程序设计称为结构化程序设计。结构化程序结构清晰，易于正确性验证，易于纠错。

下面分别对这三种基本结构进行介绍，并画出其流程图。

(1) 顺序结构。

在顺序结构中，各语句按先后顺序逐一执行。其流程图的基本形态如图 1-5 所示，语句执行顺序为：语句 1 语句 2 语句 3。

(2) 选择结构。

在选择结构中，首先对给定条件进行判断，然后根据条件的真假来确定下一步应执行哪一条语句。选择结构的基本形状有两种，如图 1-6 所示。图 1-6 (a) 的执行顺序为：当条件为真时执行语句 1，否则执行语句 2；图 1-6 (b) 的执行顺序为：当条件为真时执行语句 1，否则不执行任何语句。

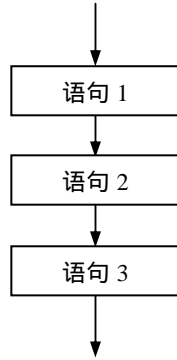


图 1-5 顺序结构的流程

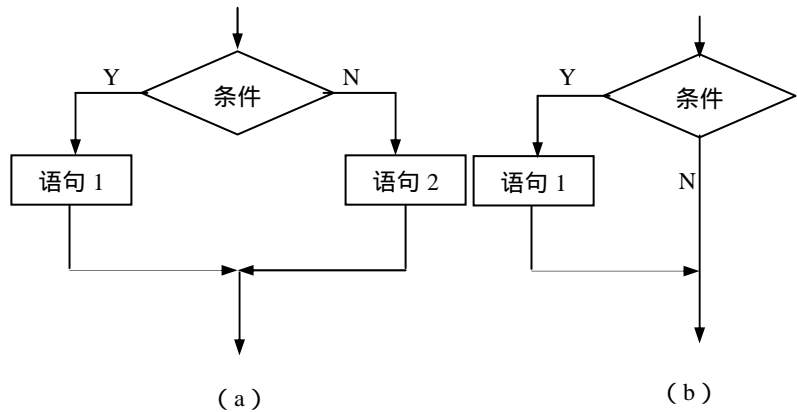


图 1-6 选择结构的流程图

(3) 循环结构。

在执行循环结构中的语句时，将根据条件，对循环结构所限定的语句（即循环体）重复执行多次或零次。

循环结构又分为“当型”循环和“直到型”循环两种情况。“当型”循环的特点是：先判断条件，当条件满足时，就执行循环体，否则就退出循环结构。“直到型”循环的特点是：先执行循环体，再判断循环条件，条件不满足时，就退出循环结构。“当型”循环结构和“直到型”循环结构的流程图分别如图 1-7 (a) (b) 所示。

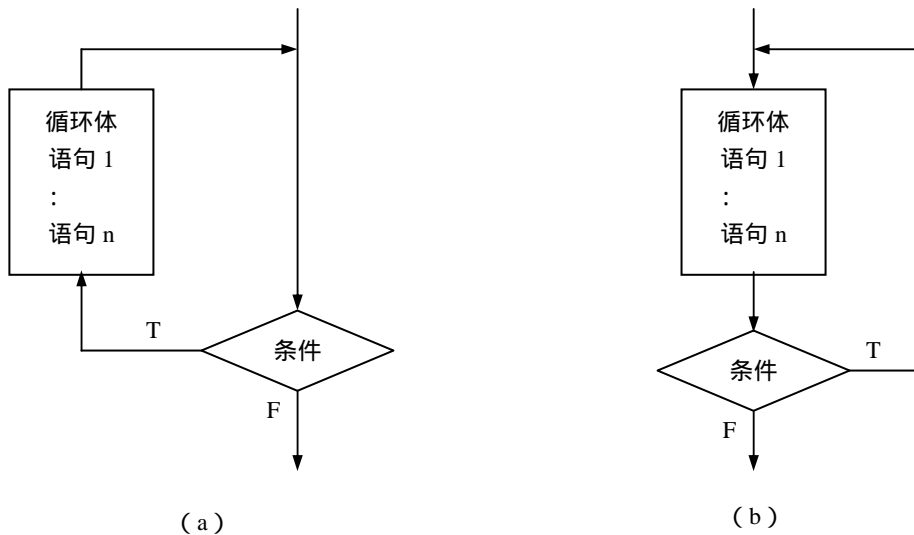


图 1-7 循环结构流程图