

国家示范性建设院校电子信息类
优质核心及精品课程规划教材

C语言程序设计

——项目教学教程

张佰慧 王德永 主编



西安电子科技大学出版社
XIDIAN UNIVERSITY PRESS

国家示范性建议院校电子信息类优质核心及精品课程规划教材

C 语言程序设计

——项目教学教程

张佰慧 王德永 主编

西安电子科技大学出版社

内 容 简 介

本书系统地介绍了C语言编程知识,全书共10章,内容包括:C语言概述、数据设计、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、模块化程序设计、指针、用户自定义数据类型和文件。

本书注重基础,突出应用,采用项目教学方法,先以任务为驱动对每个子项目进行分析讲解,再学习相关理论知识,最后进行任务实施。本书注重能力的培养,易教易学,能够学以致用,对初学者容易混淆的内容进行了重点提示和讲解。

本书可作为高职高专各相关专业的程序设计课程教材,也可作为编程开发人员培训、自学的参考书。

图书在版编目(CIP)数据

C语言程序设计——项目教学教程/张佰慧,王德永主编.

—西安:西安电子科技大学出版社,2010.9

国家示范性建议院校电子信息类优质核心及精品课程规划教材

ISBN 978-7-5606-2458-7

I. ① C… II. ① 张… ② 王… III. ① C语言—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆CIP数据核字(2010)第146330号

策 划 杨丕勇

责任编辑 宁晓蓉 杨丕勇

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西光大印务有限责任公司

版 次 2010年9月第1版 2010年9月第1次印刷

开 本 787毫米×1092毫米 1/16 印张 16.375

字 数 386千字

印 数 1~3000册

定 价 26.00元

ISBN 978-7-5606-2458-7/TP·1226

XDUP 2750001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

C 语言是一种优秀的结构化程序设计语言,其结构严谨、数据类型完整、语句简练灵活、运算符丰富。近年来许多高校把 C 语言作为理工类非计算机专业学习程序设计的第一语言和计算机专业必修的程序设计语言。C 语言教材种类繁多,而且不乏规划教材和优秀教材。但自上世纪 90 年代以来,我国的高职高专教育取得了很大的发展,为国民经济建设培养了大批急需的专门人才,高等职业技术教育呈现出勃勃生机,而与之形成鲜明对比的是适应高职高专教育水准及特色的教材编写工作相对滞后。到目前为止,全国统一的符合高职高专教育特点的基于工作过程的 C 语言规划教材比较少,因此高职高专计算机类及其他理工类专业规划教材的编写已迫在眉睫,并且十分必要。

针对项目教学的特点,本书共分为 10 章。

第 1 章 C 语言概述。本章主要介绍 C 语言出现的历史背景;C 语言的特点;C 程序基本结构和 C 程序的上机步骤。

第 2 章 “班级学生成绩管理系统”相关数据设计。本章主要介绍 C 语言的数据类型;常量和变量;整型、实型和字符型数据;各种数据类型之间的混合运算;算术运算符和算术表达式;逗号运算符和逗号表达式;关系运算符和关系表达式;逻辑运算符和逻辑表达式。

第 3 章 学生成绩输入/输出界面设计——顺序结构程序设计。本章主要介绍数据输入/输出的概念;字符输入/输出库函数;格式输入/输出库函数的格式;格式指示符的应用方法。

第 4 章 用选择语句实现对菜单的选择——选择结构程序设计。本章主要介绍关系表达式和逻辑表达式在选择结构中的应用;if 语句和 switch 语句;单分支结构和多分支结构的程序设计。

第 5 章 学生成绩统计分析——循环结构程序设计。本章主要介绍 goto 语句及用 goto 语句构成循环;while 语句、do-while 语句、for 语句;三种循环语句之间的差别;break、continue 语句;循环结构的程序设计。

第 6 章 学生成绩排名——数组。本章主要介绍一维数组、二维数组的定义和引用;字符数组的定义和引用;字符数组的输入/输出和字符串处理函数;字符串和字符串的结束标志;使用数组进行程序设计。

第 7 章 项目整体框架设计——模块化程序设计。本章主要介绍函数的定义形式;函数的参数和函数的返回值;函数的调用和嵌套调用;函数的递归调用;数组作为函数的参数;变量的存储类型;利用函数进行结构化程序设计;宏定义与文件包含的概念与应用。

第 8 章 指针。本章主要介绍地址和指针的概念;变量的指针和指向变量的指针变量;数组的指针和指向数组的指针变量;字符串的指针和指向字符串的指针变量;函数的指针

和指向函数的指针变量；返回指针值的函数；指针数组和指向指针的指针；使用指针进行程序设计。

第9章 用户自定义数据类型。本章主要介绍结构体类型的定义；结构体变量的引用和初始化；结构体数组；指向结构体类型的指针；共用体数据类型的定义和引用；枚举类型；使用结构体进行程序设计。

第10章 文件。本章主要介绍文件类型的指针；文件的基本操作；文件的打开与关闭、文件的读写；文件的定位；使用文件进行程序设计。

本书由多年从事一线计算机教学工作的教师编写，为突出高职高专学生技术应用能力的培养，营造职业氛围，缩小在校学习与生产岗位需求之间的距离，故采用了项目化的方式组织教材内容。

本书以“学生成绩管理系统”为主线，将项目细化为若干个子模块，并将每个子模块的相关知识分散到各个章节中，每个章节又以任务为驱动展开知识点。本书涵盖的知识点有：C语言的基础知识、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、模块程序设计、指针、用户自定义数据类型和文件。

本书由平顶山工业职业技术学院张佰慧、王德永担任主编。第1章、第3章由张佰慧编写，第2章由王德永编写，第4章由王聪编写，第5章、第8章由张会喜编写，第6章和第7章由杨永辉编写，第9章和第10章由张少龙编写。全书由王德永统稿。

为方便教师的教学工作，本书附有全套电子课件，需要者可从出版社网站免费下载。

由于时间仓促，加之水平有限，书中难免会有不足之处，敬请专家和读者批评指正。

编 者

2010年5月

目 录

第 1 章 C 语言概述	1
一、C 语言的发展历史及特点	1
二、C 程序的基本结构分析	3
三、程序设计时的算法描述	5
四、Visual C++ 6.0 环境下调试 C 程序实例	9
五、“班级学生成绩管理系统”初步总体规划设计	16
本章小结	19
习题	19
第 2 章 “班级学生成绩管理系统”相关数据设计	22
任务 “班级学生成绩管理系统”相关数据设计	22
一、任务情境	22
二、知识必备	22
三、任务实施	29
四、知识扩展	30
任务小结	36
习题	36
第 3 章 学生成绩输入/输出界面设计——顺序结构程序设计	38
任务一 用输入/输出函数初步设计项目封面与菜单	38
一、任务情境	38
二、知识必备	40
三、任务实施	48
四、知识扩展	51
任务小结	57
任务二 项目封面及菜单的顺序执行设计	58
一、任务情境	58
二、知识必备	58
三、任务实施	60
任务小结	64
习题	64
第 4 章 用选择语句实现对菜单的选择——选择结构程序设计	67
任务一 用 if 语句实现菜单的选择执行	67

一、任务情境.....	67
二、知识必备.....	68
三、任务实施.....	72
四、知识扩展.....	75
任务小结.....	81
任务二 用 switch 语句实现菜单的选择执行	81
一、任务情境.....	81
二、知识必备.....	81
三、任务实施.....	84
四、知识扩展.....	86
任务小结.....	91
习题.....	91
第 5 章 学生成绩统计分析——循环结构程序设计	94
任务一 用循环语句实现项目主菜单的选择执行	94
一、任务情境.....	94
二、知识必备.....	96
三、任务实施.....	101
四、知识扩展.....	104
任务小结.....	108
任务二 学生成绩统计分析——总分、平均分的计算	108
一、任务情境.....	108
二、知识必备.....	109
三、任务实施.....	112
四、知识扩展.....	113
任务小结.....	115
习题.....	116
第 6 章 学生成绩排名——数组.....	118
任务一 用数组初步实现学生最高、最低成绩查找	118
一、任务情境.....	118
二、知识必备.....	118
三、任务实施.....	121
四、知识扩展.....	123
任务小结.....	127
任务二 用数组初步实现学生成绩排序	127
一、任务情境.....	127
二、知识必备.....	127

三、任务实施.....	131
四、知识扩展.....	133
任务小结.....	136
习题.....	136
第7章 项目整体框架设计——模块化程序设计.....	139
任务 项目整体框架设计.....	139
一、任务情境.....	139
二、知识必备.....	139
三、任务实施.....	148
四、知识扩展.....	156
任务小结.....	165
习题.....	165
第8章 指针.....	169
任务一 用指针实现学生最高、最低等成绩查找.....	169
一、任务情境.....	169
二、知识必备.....	169
三、任务实施.....	177
四、知识扩展.....	179
任务小结.....	180
任务二 用指针实现学生成绩排序.....	180
一、任务情境.....	181
二、知识必备.....	181
三、任务实施.....	186
四、知识扩展.....	188
任务小结.....	194
习题.....	194
第9章 用户自定义数据类型.....	198
任务 用结构体实现学生数据的增加、删除、修改和显示.....	198
一、任务情境.....	198
二、知识必备.....	199
三、任务实施.....	206
四、知识扩展.....	213
任务小结.....	220
习题.....	220
第10章 文件.....	225
任务 学生数据的存储和重复使用.....	225

1.1.1 二、任务情境.....	225
1.1.2 二、知识必备.....	225
1.1.3 三、任务实施.....	237
1.1.4 四、知识扩展.....	239
1.2 任务小结.....	242
1.3 习题.....	242
附录 1 常用字符与 ASCII 代码对照表.....	245
附录 2 C 语言运算符的优先级和结合性.....	246
附录 3 Turbo C 常用库函数.....	247
参考文献.....	254

第 1 章 C 语言概述

知识目标:

- 了解 C 语言的特点、C 语言程序开发步骤。
- 初步了解 C 程序的组成结构、主函数的作用。
- 初步掌握 C 语言流程图、N-S 图图例特点与属性。

能力目标:

- 能够启动 Visual C++ 6.0, 并能正确进入编程窗口。
- 学会与人打交道, 完成任务调查。
- 能初步掌握“班级学生成绩管理系统”工作模块构成。
- 能调查了解本校学生成绩管理系统的工作流程, 画出本校学生成绩管理系统工作模块图。

一、C 语言的发展历史及特点

1. C 语言的发展历史

C 语言是目前世界上较为流行、使用非常广泛的高级程序设计语言。对于操作系统和系统应用程序以及需要对硬件进行操作的场合, C 语言明显优于其它高级语言, 许多大型应用软件都是用 C 语言编写的。C 语言具有强大的绘图能力, 可移植性好, 并具备很强的数据处理能力, 因此适于编写系统软件, 同时它也可用于数值计算。

C 语言的原型是 ALGOL 60 语言(也称 A 语言)。

1963 年, 剑桥大学将 ALGOL 60 语言发展成为 CPL(Combined Programming Language)语言。1967 年, 剑桥大学的 Martin Richards 对 CPL 语言进行了简化, 产生了 BCPL 语言。

1970 年, 美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改, 并为它起了一个有趣的名字“B 语言”。意思是将 CPL 语言煮干, 提炼出它的精华, 他用 B 语言写了第一个 UNIX 操作系统。而在 1973 年, B 语言也给人“煮”了一下, 美国贝尔实验室的 Dennis.M.Ritchie 在 B 语言的基础上最终设计出了一种新的语言, 他取了 BCPL 的第二字母作为这种语言的名字, 这就是 C 语言。为了使 UNIX 操作系统得以推广, 1977 年 Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年 Brian W.Kernighan 和 Dennis M.Ritchie 出版了著名的《The C Programming Language》一书, 从而奠定了 C 语言成为目前世界上最广泛流行的高级程序设计语言的基础。

1988 年, 随着微型计算机的日益普及, 出现了许多 C 语言版本。由于没有统一的标准, 使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况, 美国国家标准学会(ANSI)为 C 语言制定了一套 ANSI 标准, 成为现行的 C 语言标准。

C 语言发展迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。主流的三种操作系统 Windows、Linux、UNIX 其内核都是用 C 语言和汇编语言编写的,几乎所有的网络游戏、百度搜索引擎(Baidu.com)等许多应用软件也是用 C 语言编写的。

目前广泛使用的 C 语言编译器有以下几种:

- Microsoft C 或称 MS C
- Borland Turbo C 或称 Turbo C
- Win-TC

这些 C 语言版本不仅实现了 ANSI C 标准,而且在此基础上各自作了一些扩充,使之更加方便、完善。

在 C 的基础上,1983 年贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言,成为一种面向对象的程序设计语言。C++ 目前流行的最新版本是 Borland C++ 6.0、Digital Mars C++ 8.50 和 Microsoft Visual C++ 2008。C++ 提出了一些更为深入的概念,它所支持的面向对象的概念容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法,因而也增加了整个语言的复杂性,掌握起来有一定难度。

2. C 语言的特点

一种语言之所以能存在和发展,并且具有较强的生命力,总是有其不同于其他语言的特点。C 语言的主要特点如下:

1) 简洁紧凑,灵活方便

标准的 C 语言一共只有 32 个关键字,9 种控制语句,程序书写自由。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元。

2) 运算符丰富

C 语言的运算符包含的范围很广泛,共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算处理,从而使 C 语言的运算类型极其丰富,表达式类型多样。灵活使用各种运算符可以实现在其它高级语言中难以实现的运算。

3) 数据结构丰富

C 语言的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能用来实现各种复杂数据结构的运算,并引入了指针概念,使程序效率更高。另外,C 语言具有强大的图形功能,支持多种显示器和驱动器,且计算功能、逻辑判断功能强大。

4) C 语言是结构式语言

结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言是以函数形式提供给用户的,这些函数可方便地调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

5) C 语言的语法限制不太严格,程序设计自由度大

一般的高级语言语法检查比较严格,能够检查出几乎所有的语法错误。而 C 语言允许

程序编写者有较大的自由度。

6) C 语言允许直接访问物理地址, 可以直接对硬件进行操作

C 语言既具有高级语言的功能, 又具有低级语言的许多功能, 能够像汇编语言一样对位、字节和地址进行操作, 可以用来编写系统软件。

7) C 语言程序生成代码质量高, 程序执行效率高

C 语言生成的代码一般只比汇编程序生成的目标代码效率低 10%~20%。

8) C 语言适用范围广, 可移植性好

C 语言适合于多种操作系统, 如 Windows、DOS、UNIX, 也适用于多种机型。

二、C 程序的基本结构分析

为了说明 C 语言源程序结构的特点, 先看以下几个程序。这几个程序由简到难, 表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍, 但可从这些例子中了解到一个 C 语言源程序的基本组成部分和书写格式。

【例 1.1】 一个最简单的 C 语言程序。

程序代码如下:

```
01 #include<stdio.h>
02 void main()
03 {
04     printf("Good Moning!\n");
05 }
```



程序第 02 行 main 是主函数的函数名, 表示这是一个主函数。每一个 C 源程序都必须有且只能有一个主函数(main 函数)。第 04 行是函数调用语句, printf 函数的功能是把要输出的内容送到显示器去显示。printf 函数是一个由系统定义的标准函数, 可在程序中直接调用。

例 1.1 是一个最简单的 C 语言程序。main 前面的 void 表示此主函数是“空类型”, void 是“空”的意思, 即执行此函数后不产生一个函数值。每一个 C 语言程序都必须有一个 main 函数。每一个函数要有函数名, 也要有函数体(即函数的实体)。函数体由一对花括号 { } 括起来。本例中主函数内只有一行。printf 是 C 编译系统提供的标准函数库中的输出函数(详见第 4 章)。程序第 04 行是一个 printf 语句, 圆括号中引号内的字符串按原样输出。“\n”是换行符, 在执行程序时, 输出“Good Moning!”, 然后执行回车换行。语句最后有一个分号。

在使用标准函数库中的输入/输出函数时, 编译系统要求程序提供有关的信息(例如对这些输入/输出函数的声明), 程序第 01 行“#include<stdio.h>”的作用就是用来提供这些信息的。stdio.h 是 C 编译系统提供的一个文件名, stdio 是“standard input & output”的缩写, 即有关“标准输入/输出”的信息。开始时对此可暂不必深究, 以后会有详细介绍。在此只需记住: 在程序中用到系统提供的标准函数库中的输入/输出函数时, 应在程序的开头写这样一行:

```
#include<stdio.h>
```

【例 1.2】 任意输入两个数, 求两个数中较大者。

程序代码如下:

```
01 #include<stdio.h>
02 main()
03 { int x,y,z; /*变量说明*/
04     int max(int a,int b); /*函数声明*/
05     printf("input two numbers:\n");
06     scanf("%d%d",&x,&y); /*输入 x、y 值*/
07     z=max(x,y); /*调用 max 函数*/
08     printf("max=%d\n",z); /*输出*/
09 }
10 int max(int a,int b) /*定义 max 函数*/
11 {
12     if(a>b)
13         return a; /*把结果返回主调函数*/
14     else
15         return b; /*把结果返回主调函数*/
16 }
```



程序中第 03 行是变量说明,说明变量 x、y、z 的数据类型。程序第 04 行是函数声明。第 06 行为输入语句,调用 scanf 函数,接收键盘上输入的数并存入变量 x、y 中。程序第 07 行调用了 max 函数。程序第 10~16 行是一个用户自定义的 max 函数。

例 1.2 中程序的功能是,由用户输入两个整数,程序执行后输出其中较大者。本程序由两个函数组成,主函数 main 和 max 函数,函数之间是并列关系,但可从主函数中调用其它函数。本例中的主函数体又分为两部分:说明部分(第 03、04 行)和执行部分(第 05~08 行)。在程序的说明部分中,不仅可以有变量说明,还可以有函数说明。

max 函数的功能是比较两个数,然后把较大的数返回给主函数。程序第 04 行是在主函数中对被调用函数 max 的声明。由于在主函数中要调用 max 函数,而 max 函数的定义却在 main 函数之后,为了使编译系统能够正确识别和调用 max 函数,必须在调用 max 函数之前对 max 函数进行声明,以通知编译系统:“在 main 函数中, max 是一个函数名”。关于函数的详细内容将在第 7 章介绍。程序的每行后用“/*”和“*/”括起来的内容为注释部分,程序不执行注释部分。

上例中程序的执行过程是,首先在屏幕上显示提示串,请用户输入两个数,回车后由 scanf 函数语句接收这两个数送入变量 x、y 中,然后调用 max 函数,并把 x、y 的值(称为实际参数)传送给 max 函数的参数 a、b(称为形式参数)。在 max 函数中比较 a、b 的大小,把大者返回给主函数的变量 z,最后在屏幕上输出 z 的值。

程序运行结果如下:

input two numbers:

8, 5 ✓

max=8

为了在分析运行结果时便于区别输入和输出的信息，本书对输入的信息加了下划线，如上面的运行结果的第2行表示：从键盘输入8和5，然后按Enter键。第3行是从计算机输出的信息，显示在屏幕上。

本例用到了函数调用、实际参数和形式参数等概念，在此只做了很简单的解释。读者如对此不大理解，可以先不予以深究，在以后的章节中会学到这部分知识。在此介绍该例，主要为使读者对C程序的组成和形式有一个初步的了解。

通过以上几个例子，可以看到C源程序如下的结构特点：

- (1) 一个C语言源程序可以由一个或多个源文件组成。
- (2) 每个源文件可由一个或多个函数组成。
- (3) 一个源程序不论由多少个函数组成，都有一个且只能有一个main函数，即主函数。
- (4) 源程序中可以有预处理命令(include命令仅为其中的一种)，预处理命令通常应放在源文件或源程序的最前面。
- (5) 每一个说明、每一条语句都必须以分号结尾。但预处理命令、函数头和花括号“{”之后不能加分号。
- (6) 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

三、程序设计时的算法描述

1. 程序的三种基本结构

一个程序包含一系列指令，每一条指令使计算机完成一种操作。程序中的指令不是任意书写而无规律的。1966年，Bohra和Jacopini提出了三种基本结构，用这三种基本结构作为一个良好算法的基本单元，任何一个复杂程序都可由这三种基本结构组成，即顺序结构、选择结构和循序结构。

1) 顺序结构

在顺序结构中程序是一条语句接一条语句顺序地往下执行的。顺序结构的程序是最简单的程序。图1-1为顺序结构示意图。

2) 选择结构

在程序执行过程中，程序的流程由多路分支组成，根据不同的条件去执行不同的任务。图1-2为选择结构示意图。

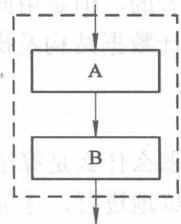


图 1-1 顺序结构

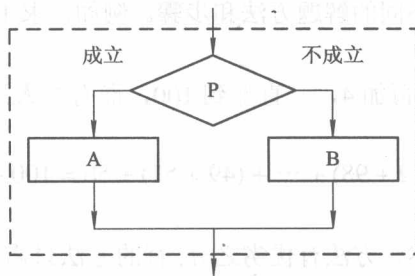


图 1-2 选择结构

3) 循环结构

若在程序中需要根据某项条件重复地执行某项任务若干次，或直到满足或不满足某条件为止，这就构成了循环结构。图 1-3 为循环结构示意图。

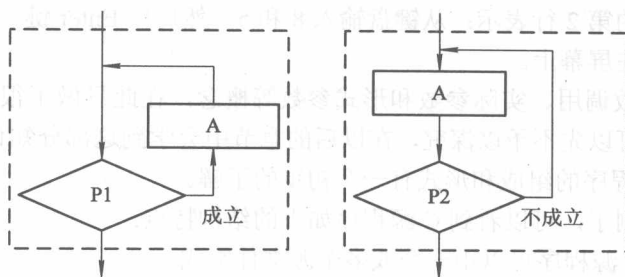


图 1-3 循环结构

这三种结构是程序设计中最基本的问题，也是程序设计的基础。

2. 程序的算法描述

1) 算法的概念

计算机所进行的一切操作都是由程序决定的，而程序是程序员事先编写好并输入计算机的。从前面的程序中可知，一个程序包括以下两个方面的内容：

(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，即数据结构(data structure)。

(2) 对操作的描述，即操作步骤，也就是算法(algorithm)。

数据是操作的对象，操作的目的是对数据进行加工处理，以得到期望的结果。例如，厨师制作菜肴，需要有菜谱，菜谱一般应包括：① 配料，指出应使用哪些原料；② 操作步骤，指出如何使用这些原料按规定的步骤加工成所需的菜肴。没有原料是无法加工成所需菜肴的，但是用同一些原料可以加工出不同的菜肴，作为程序设计人员，必须认真考虑和设计数据结构及操作步骤(即算法)。著名计算机科学家沃思(Niklaus Wirth)提出了一个公式：

$$\text{数据结构} + \text{算法} = \text{程序}$$

那么什么是算法呢？我们从事各种工作和活动，都必须事先想好进行的步骤，然后按部就班地进行，才能避免产生错乱。广义地说，为解决一个问题而采取的方法和步骤，就称为“算法”。

对同一个问题，可以有不同的解题方法和步骤。例如，求 $1+2+3+\cdots+100$ ，即 $\sum_{n=1}^{100} n$ ，有

人可能先进行 $1+2$ ，再加 3，再加 4，一直加到 100，而有的人采取这样的方法：

$$\sum_{n=1}^{100} n = 100 + (1+99) + (2+98) + \cdots + (49+51) + 50 = 100 + 49 \times 100 + 50 = 5050$$

还可以有其他方法。当然，方法有优劣之分，有的方法只需很少的步骤，而有些方法则需要较多的步骤。一般来说，希望采用简单、运算步骤少的方法。因此，为了有效地进行

解题，不仅需要保证算法正确，还要考虑算法的质量，选择合适的算法。

2) 算法的表示

构思好一个算法后，可以选择不同的方式表示。

(1) 用自然语言表示。

自然语言就是人们日常使用的语言，可以是汉语、英语或其他语言。用自然语言表示通俗易懂，但文字冗长，容易出现歧义。自然语言表示的含义往往不太严格，要根据上下文才能判断其正确含义。假如有这样一句话：“张先生对李先生说他的孩子考上了大学”，请问是张先生的孩子考上了大学还是李先生的孩子考上了大学呢？仅仅从这句话本身难以判断。此外，用自然语言来描述包含分支和循环的算法不很方便。因此，除了那些很简单的问题以外，一般不用自然语言描述算法。

(2) 用传统流程图表示。

流程图是用一些图框来表示各种操作。用图形表示算法直观形象，易于理解。美国国家标准化学会 ANSI 规定了一些常用的流程图符号，如图 1-4 所示，已为世界各国程序工作者普遍采用。

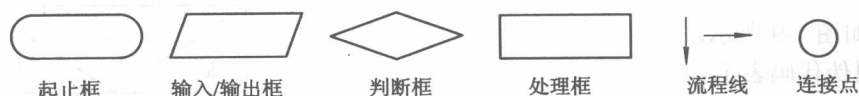


图 1-4 流程图的常用符号

【例 1.3】 任意输入两个数，求两个数中的较大者。

用传统流程图表示此题的算法如图 1-5 所示。

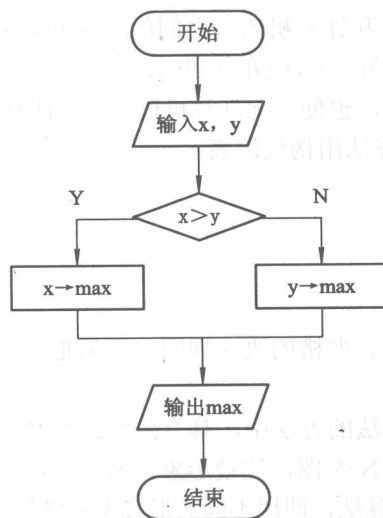


图 1-5 求两个数中较大者的传统流程图

(3) 用 N-S 流程图表示。

1973 年美国学者 I.Nassi 和 B.Shneiderman 提出了一种新的流程图形式。在这种流程图中，完全去掉了带箭头的流程线。全部算法写在一个矩形框内，在该框内还可以包含一些从属于它的小矩形框，或者说，由一些基本的框组成一个大的框。这种流程图又称 N-S 结

结构化流程图(N 和 S 是两位美国学者英文姓氏的首字母)。这种流程图适于结构化程序设计,而且作图简单,占用面积小。三种基本程序结构的 N-S 流程图如图 1-6~图 1-8 所示。

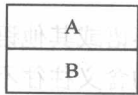


图 1-6 顺序结构

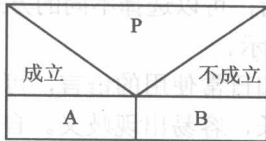


图 1-7 选择结构

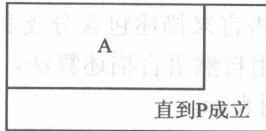
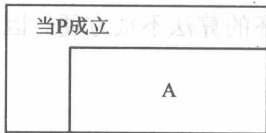


图 1-8 循环结构

【例 1.4】 将例 1.3 的算法用 N-S 流程图来表示。

结果如图 1-9 所示。

(4) 用伪代码表示。

用传统流程图和 N-S 图表示算法直观易懂,但画起来比较费事,在设计一个算法时,可能要反复修改,而修改流程图是比较麻烦的。为了便于设计算法,常用一种称为伪代码(pseudo code)的工具。

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。它如同一篇文章一样,自上而下地写下来。每一行(或几行)表示一个基本操作。它不用图形符号,因此书写方便、格式紧凑、比较好懂,也便于向计算机语言算法(即程序)过渡。

【例 1.5】 将例 1.3 的算法用伪代码表示。

```
if x>y then
    x→max
else
    y→max
```

用伪代码写算法并无固定、严格的语法规则,只需把意思表达清楚,并且书写的格式清晰易读即可。

总之,在以上几种表示算法的方法中,具有熟练编程经验的专业人士喜欢用伪代码,初学者则喜欢用传统流程图或 N-S 图,比较形象,易于理解。

至今为止,我们只是描述算法,即用不同的形式表示操作的步骤,而要得到运算结果,就必须实现算法。我们的任务是用计算机解题,也就是要用计算机实现算法。计算机是无法识别流程图和伪代码的,也就是说用流程图和伪代码描述的算法是无法在计算机上实现的,只有用计算机语言编写的程序才能被计算机执行(当然还要编译成目标程序才能被计算机识别和执行)。因此,在用流程图或伪代码描述出一个算法后,还要将它转换成计算机语言程序。

用计算机语言表示算法必须严格遵循所用语言的语法规则,这是和伪代码不同的。

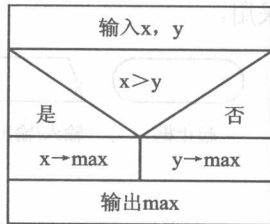


图 1-9 求两个数中较大者的 N-S 流程图