

010010010101001010110101

武汉大学出版社

计算机科学与技术系列教材

C语言程序设计教程

主 编 陈建新 李 勇 陈佛敏
副主编 谭成予 焦家林 陈晓文



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

C 语言程序设计教程/陈建新,李勇,陈佛敏主编. —武汉:武汉大学出版社,2007.8

计算机科学与技术系列教材

ISBN 978-7-307-05652-7

I. C… II. ①陈… ②李… ③陈… III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 084157 号

责任编辑:林 莉 责任校对:王 建 版式设计:支 笛

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:wdp4@whu.edu.cn 网址:www.wdp.com.cn)

印刷:湖北金海印务公司

开本:787×1092 1/16 印张:13.125 字数:325 千字

版次:2007 年 8 月第 1 版 2007 年 8 月第 1 次印刷

ISBN 978-7-307-05652-7/TP·251 定价:22.00 元

版权所有,不得翻印;凡购买我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。

内 容 简 介

C 语言是目前最流行的程序设计语言之一，具有概念简洁、语句简凑、表达能力强、运算符多而灵活、控制流和数据结构新颖，程序结构性和可读性好，可移植性好等优点。C 语言数据类型丰富，既可以用来编写应用程序，又可以用来编写系统软件。它既具有高级语言程序设计的特点，又具有汇编语言的功能，是当今世界上最具影响力的程序设计语言之一。

本书全面介绍了 C 语言程序设计的基本知识，C 语言的基本数据类型、常量、变量与表达式，C 语言进行结构化程序设计的基本方法，包括结构化程序的顺序结构、选择结构、循环结构及其设计方法，介绍了函数与指针的概念及其使用，还介绍了其他数据类型与 C 语言文件系统等内容。

本书可作为高等院校开设“C 语言程序设计”课程的教学用书，也可作为从事计算机应用的科技人员的参考用书和培训教材，同时也可作为参加全国计算机等级考试者的参考用书。



前 言

C 语言是目前最流行的程序设计语言之一,具有概念简洁、语句简凑、表达能力强、运算符多而灵活、控制流和数据结构新颖、持续结构性和可读性好、可移植性好等优点。有利于培养良好的编程风格,且易于体现结构化程序设计的思想。C 语言数据类型丰富,既可以用来编写应用程序,又可以用来编写应用软件。它既具有高级语言程序设计的特点,又具有汇编语言的功能,是当今世界上最具影响力的程序设计语言之一,也是程序设计工作者必须熟练掌握的一种语言工具。高等院校几乎所有的理工科专业都开设了 C 语言程序设计课程。C 语言作为学习程序设计的入门课程,它又是学习计算机其他有关课程的基础,因此学好 C 语言成为一种越来越普遍的需要。

本书全面介绍了 C 语言的概念、特性和结构化程序设计的方法。从 C 语言的发展与特点出发,系统地介绍了 C 语言程序设计中的变量、运算符、表达式、数据类型、存储类别、语句、函数、指针、结构体与共用体、文件等。本教材始终以程序设计为主线,注重培养学生程序设计的思维方式和技巧。每章都配有习题,作为学生对该章内容学习的巩固和延伸。

全书共分八章,每章开头都给出了“学习目标”,读者通过阅读“学习目标”,明确本章的学习目的,需要掌握的内容,以增强学习的主动性和积极性。在第一章,介绍了 C 语言程序设计的基本知识。第二章介绍了 C 语言的基本数据类型、常量、变量与表达式。第三、四章介绍了 C 语言进行结构化程序设计的基本方法,包括结构化程序的顺序结构、选择结构、循环结构及其设计方法。第五、六章对 C 语言的精华部分——函数与指针的内容作了详细的阐述。第七章介绍了其他数据类型,如结构体类型的定义与引用方法,用结构体进行链表的简单操作,共用体及枚举类型的概念、定义和引用,使用 typedef 定义类型别名的方法。第八章介绍了文件操作的基本方法,文本文件与二进制文件读写函数的使用以及顺序文件和随机文件的创建与使用的方法。

本书是作者在多年从事 C 语言及计算机专业相关课程的教学的基础上总结经验编写而成的。在编写过程中还参考了相关优质课程网站上的内容,内容充实,在选材上注重系统性、适用性和先进性。全书所选例子,均已上机通过。

本书由陈建新、李勇、陈佛敏、谭成予、焦家林、陈晓文编写,陈建新负责全书的统稿。

在本书的编写过程中,得到了武汉大学计算机学院、孝感学院计算机系、咸宁学院计算机系、襄樊学院计算机系的领导及老师们的大力支持和帮助,武汉大学出版社为本书的出版做了大量的工作,在此一并表示衷心的感谢。同时也对编者在编写本书的过程中参考过的相关教材的作者表示感谢。

本书可作为高等院校开设“C 语言程序设计”课程的教学用书,也可作为从事计算机应用的科技人员的参考用书和培训教材,同时还可作为参加全国计算机等级考试者的参考用书。

由于作者水平有限,书中难免有错误和疏漏之处,敬请读者批评指正。

编 者

2007 年 4 月

目 录

第一章 C 语言概述.....	1
1.1 绪论	1
1.1.1 程序设计语言	1
1.1.2 C 语言简介	2
1.2 C 语言程序的构成	2
1.2.1 C 语言的字符集	2
1.2.2 C 程序的基本词汇	3
1.2.3 C 语言的基本语句	3
1.2.4 C 程序的基本结构	4
1.3 简单的 C 语言源程序	4
1.4 C 语言实验环境——Turbo C 操作界面	5
1.5 算法描述工具简介	6
1.5.1 程序流程图	7
1.5.2 N-S 图	8
1.5.3 伪代码	8
习题一	9
第二章 C 程序设计基础	10
2.1 基本数据类型	10
2.2 常量与变量	11
2.2.1 整型量	11
2.2.2 浮点型量	12
2.2.3 字符型量	13
2.3 数组	17
2.3.1 一维数组	17
2.3.2 二维数组	19
2.3.3 字符数组与字符串	20
2.4 常用运算符及其表达式	23
2.4.1 算术运算符	23
2.4.2 赋值运算符	24
2.4.3 关系运算符	26
2.4.4 逻辑运算符	26
2.4.5 位运算符	27



2.4.6 其他运算符	28
2.4.7 运算符的优先级与结合性	29
2.5 基本输入输出函数	30
2.5.1 printf()函数	30
2.5.2 scanf()函数	32
2.5.3 字符的输入输出函数	35
2.6 预处理命令	35
习题二	41
第三章 顺序与分支程序设计	43
3.1 C 语言语句及其程序结构	43
3.1.1 流程控制语句	43
3.1.2 表达式语句	43
3.1.3 特殊语句	44
3.1.4 C 程序的基本结构	45
3.2 顺序程序设计及其举例	45
3.3 分支语句	48
3.3.1 逻辑表达式	48
3.3.2 单分支 if 语句	49
3.3.3 双分支 if 语句	50
3.3.4 多分支 if 语句	52
3.3.5 switch 语句	53
3.4 goto 语句	55
3.5 分支程序设计举例	55
习题三	62
第四章 循环结构程序设计	65
4.1 while 语句	65
4.2 do-while 语句	67
4.3 for 语句	68
4.4 三种循环的比较	71
4.5 终止循环语句	71
4.6 return 语句	73
4.7 多重循环	73
4.8 循环程序设计举例	76
习题四	86
第五章 函数	88
5.1 函数的定义	88
5.1.1 库函数	88



5.1.2 用户自定义函数	92
5.2 函数的调用及参数传递	94
5.2.1 函数的说明	94
5.2.2 函数的调用	95
5.2.3 函数的参数传递	96
5.3 函数的嵌套调用及递归调用	97
5.3.1 嵌套调用	97
5.3.2 递归调用	98
5.4 函数与数组	101
5.4.1 数组元素作函数实参	101
5.4.2 数组名作函数参数	101
5.5 变量的作用域及存储类型	104
5.5.1 局部变量与全局变量	104
5.5.2 变量的存储类型	107
5.5.3 函数的存储分类	110
5.6 函数应用举例	111
习题五	114
第六章 指针	117
6.1 指针的概念	117
6.2 指针运算及多级指针	119
6.2.1 与指针相关的运算符	119
6.2.2 指针运算	119
6.2.3 多级指针	122
6.3 指针与数组的应用	123
6.3.1 指向一维数组元素的指针	123
6.3.2 二维数组及其指针表示	124
6.3.3 指针数组	128
6.3.4 字符串与指针	129
6.3.5 main 函数的参数	130
6.4 指针与函数	131
6.4.1 指针变量作为函数参数	131
6.4.2 指针型函数	133
6.4.3 函数指针	134
6.5 指针应用举例	134
习题六	136
第七章 其他数据类型	141
7.1 结构体类型	141
7.1.1 结构体类型的定义	141



7.1.2 结构体变量的定义	142
7.1.3 结构体变量的引用	144
7.1.4 结构体变量的初始化	144
7.1.5 结构体作函数参数	146
7.1.6 结构体的嵌套	147
7.2 结构体数组	147
7.3 结构体类型指针	150
7.3.1 指向结构体变量的指针	150
7.3.2 指向结构体数组的指针	150
7.3.3 结构体与函数	151
7.3.4 结构型函数和结构指针型函数	152
7.4 动态数据结构——链表	153
7.4.1 链表的结构描述	154
7.4.2 单链表的建立	155
7.4.3 单链表的基本操作	157
7.5 位段	162
7.6 共用体类型	164
7.6.1 共用体类型定义	164
7.6.2 共用体变量的定义	164
7.6.3 共用体变量的引用	165
7.7 枚举类型与自定义类型	167
7.7.1 枚举类型的定义	167
7.7.2 枚举类型变量的定义	168
7.7.3 用 typedef 定义类型	170
7.7.4 typedef 类型定义的使用	171
7.8 程序设计举例	171
习题七	174
第八章 文件	177
8.1 文件概述	177
8.1.1 文件的相关概念	177
8.1.2 文件系统	178
8.1.3 文件指针	179
8.2 文件的打开与关闭	179
8.2.1 文件的打开	180
8.2.2 文件的关闭	181
8.3 文件的读与写	181
8.3.1 字符读写函数	182
8.3.2 字符串读写函数	184
8.3.3 数据块读写函数	185



8.3.4 格式化读写函数	187
8.4 其他文件处理函数	188
8.5 文件应用举例	190
习题八	192
主要参考文献	193



第一章 C语言概述



〔学习内容及要求〕本章是C语言的基础,主要介绍C语言的基本语法。万事开头难,当我们第一次涉及一些新的概念和名词时,的确容易被一些抽象的东西所迷惑。本章只要求学生理解一些初步的程序设计知识,如常量、变量、数据类型等概念,并简单了解C语言程序的结构、程序运行过程及算法的描述方法,熟悉C语言的开发环境。

重点:C语言源程序的基本结构。

难点:Turbo C工作环境的使用及对源程序的调试与运行。

1.1 绪 论

1.1.1 程序设计语言

从计算机诞生以来,就有程序,程序是对解决应用问题的方法的代码描述;而站在计算机的角度上看,程序是用某种计算机能理解并执行的计算机语言作为描述语言,对解决问题的方法步骤的描述。计算机执行按程序所描述的方法步骤,能完成特定的功能。

程序语言有多种,计算机能识别和理解的语言与人类所理解的语言相差很大。伴随着计算机技术的发展,程序设计语言经过了四代。

1. 机器语言

由计算机能直接理解和执行的二进制代码组成的程序语言称为机器语言。这种语言中指令和数据都用二进制方式表示,机器语言程序可以不经处理直接由机器执行,但这种语言书写、理解、修改都十分困难,程序编写效率低,且易出错。

2. 汇编语言

用有助于记忆的英语符号、单词缩写来代表二进制代码,这样的程序语言称为汇编语言。汇编语言与机器语言几乎是一一对应的,用汇编语言编写的程序称为“汇编语言源程序”,汇编语言源程序不能在计算机上直接执行,需要用汇编程序将汇编源程序翻译成机器语言程序,然后执行汇编后得到的机器语言程序。机器语言和汇编语言都是低级语言,它与具体计算机硬件紧密相关,所以又称它们是面向机器的语言。

3. 面向过程的高级语言

与机器语言和汇编语言相比较,高级语言与具体计算机无关,是一种用近似于人类自然语言(主要是英语)描述算法过程的计算机程序设计语言。高级语言种类有很多,C语言就是高级语言的一种。高级语言程序主要是描述一个求解问题的解题过程,而不必考虑或很少考虑机器本身硬件特点,所以也称这类语言为面向过程语言。

用高级语言编写的程序称为“源程序”。计算机不能直接执行源程序的语句,必须通过一些软件将源程序翻译为机器能识别的目标程序,然后让机器执行。翻译的方式有解释方式和编译方式两种。



解释方式的翻译, 翻译和执行是同时进行的, 解释程序逐句取出源程序中的语句, 对它进行解释并执行, 边解释边执行, 程序解释完成则程序执行结束, 得到结果。

编译方式的翻译, 翻译和执行是分离的, 先运行编译程序, 将源程序一次性翻译产生计算机可直接执行的二进制代码程序 (称为目标程序), 然后让计算机执行目标程序得到结果。

解释方式的主要优点是计算机与人的交互性好, 适合调试程序, 边执行边改错, 较快得到一个正确的程序。缺点是逐句解释执行, 运行速度慢, 多次运行就必须多次解释。

编译方式是由编译程序将源程序翻译为目标程序, 再由机器执行目标程序。优点是一次翻译多次运行, 程序运行速度快。

C 语言是一种编译型语言。

4. 非过程化的高级语言

过程化语言中求解问题时程序员不仅要考虑“做什么”, 而且要考虑“怎么做”, 这将会使得程序员把主要精力放在具体解题步骤的设计上。非过程化的高级语言又称为第四代语言, 这种非过程化的高级语言把求解问题的重点放在“做什么”上, 程序员只需要向计算机说明做什么, 就可以由计算机自动生成和安排执行的步骤, 因此程序设计的效率很高。

1.1.2 C 语言简介

1. C 语言发展简史

1963 年, 参照 ALGOL 60 语言, 产生了一种增添了能直接存取计算机硬件的特性的高级语言, 称为 CPL 语言。1967 年, CPL 进行了进一步的简化, 改称 BCPL 语言。1970 年, BCPL 突出了硬件处理能力从而成为 B 语言, 并用于编写 UNIX 操作系统。1972 年, B 语言进一步扩充数据类型和恢复通用性, 正式命名为 C 语言, 并用 C 语言重写了 UNIX 操作系统。以后 C 语言开始流行, 为统一版本, ANSI 于 1987 年制定了 C 语言的标准, 称为 ANSI C。目前 C 语言的版本有许多, 开发环境也各有不同。

2. C 语言的主要特点

C 语言主要有以下特点:

(1) C 语言是一种高级语言, 同时也提供了较多可以直接存取硬件的手段, 与机器语言接近。所以 C 语言可以写出一些需要对硬件进行操作的程序, 并能方便地与汇编语言等低级语言混合编程。

(2) 数据类型与运算符丰富, 描述算法更简单方便。

(3) C 程序的结构简单, 语言包含的语句类别少。

(4) C 语言是一种结构化语言, 提供完善的结构化程序控制结构, 适宜采用结构化程序设计方法开发程序。

(5) 通常 C 系统都提供大量的库函数供程序开发人员选用, 能简化程序开发。

1.2 C 语言程序的构成

一个 C 语言程序是由一系列的符号构成的, 符号构成词, 词构成语句, 语句构成程序, 因此首先讨论 C 语言的字符集, 即可以出现在 C 程序中的符号。

1.2.1 C 语言的字符集

C 语言的基本字符集有:



(1) 数字符号 10 个, 即 0~9。

(2) 英文字母, 大写和小写英文字母各 26 个。

(3) 键盘符号 33 个, 用于运算符号、标点符号及其他用途的特殊符号。

(4) 转义字符, 转义字符是由字符 (\) 开始后跟单个字符或若干字符组成, 通常用于表示控制代码或特殊符号, 控制符号是不能从键盘上输入的, 所以程序中的这些符号只能通过转义的方式实现。

1.2.2 C 程序的基本词汇

程序中的词汇是指由字符构成并用空格换行等分隔符构成的连续的符号串 (这里的词汇并不仅指英语词汇)。C 语言的基本词汇有:

(1) 常量: 如 100、15.0、'A'、"ABC"。

(2) 运算符: 如 +、-、*、/、%、&&、||、!、>、>=、<、<=、==、!=、+= 等, 各种运算符的具体功能和语法在第二章学习。

(3) 保留字: 又称关键字, 它是在程序或语句中用来表示 C 语言特定语法含义的英文单词或单词的缩写, 它们在 C 语言中已赋予一定的意义, 所以其意义无需在程序中进一步解释, 因此保留字不能再作其他用途。C 语言的保留字共有 32 个:

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	int	long
register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void
volatile	while				

(4) 标识符: 用于为程序中的变量、常量、函数、标号等命名的词。在 C 语言中, 一个正确的标识符应该遵循这样的规定: 由英文字母或下画线开始, 后跟 0 个或多个字母、下画线和数字组成。

利用基本词汇, 按照给定的 C 语言的语法规则就可命名程序对象, 描述表达式计算, 构造语句、函数, 直至写出整个程序。

1.2.3 C 语言的基本语句

C 语言的语句主要分以下几种:

(1) 数据定义语句——用来定义程序中使用的各种数据, 及能存放数据的对象的名称和特性。

(2) 表达式语句——在任何有意义的表达式之后接上分号便构成语句。最常见的有赋值表达式和函数调用表达式后加分号构成的表达式语句, 分别称为赋值语句和函数调用语句。

(3) 流程控制语句——用来控制程序执行过程的语句。它们有选择控制语句、循环控制语句、break 语句、continue 语句、return 语句和 goto 语句等。

(4) 复合语句——用花括号括住的语句序列, 即构成复合语句。复合语句将其中的一系列语句组织为一个整体。

(5) 空语句——只有单个分号构成的语句。空语句表示没有任何操作, 用于选择控制或循环控制没有特别被控制的成分语句, 或在复合语句的末尾放置语句标号等。

(6) 其他语句——如类型定义语句等。



1.2.4 C 程序的基本结构

如果仅从语法上考虑, 以下就是一个最简单的 C 语言程序:

```
main()
{
}
```

通常一个 C 程序包括一个或多个函数, 这里的函数与数学中的函数的概念相似, 那就是给定一个数据, 用函数计算可以得到函数值。C 语言中的函数有两种, 一种是库函数, 它们是由系统定义的, 在程序中可以直接使用这些函数; 另一类函数是由程序员设计的。

C 语言中的函数的一般格式为:

```
返回值类型    函数名 ( 参数 )
{ 函数体      }
```

C 语言程序就是由若干个这样的函数组成的。在一个程序中, main 函数为主函数。C 程序总是从主函数开始执行。

1.3 简单的 C 语言源程序

设计程序的目的是为了解决某些问题, 而解决问题的过程大致上可以分为输入数据、处理数据和输出结果三个阶段。

例 1-1 从键盘上输入一个整数, 输出这个整数的平方值。

```
void main()
{
    int x,y;
    scanf ("%d",&x);    /*从键盘上输入一个整数并存放在变量 x 中*/
    y=x*x;
    printf ("y=%d", y);
}
```

这是一个完整的 C 语言程序, 它虽然功能简单, 却体现了 C 语言编程的一般规则。

C 语言程序的主要规则如下:

- (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 C 语言程序不论由多少个文件和函数构成, 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、每一个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。
- (7) 程序中的 {}、()、" "、' ' 都必须成对出现 (除非在一对 " " 中以字符串方式出现)。
- (8) C 语言中严格区分大小写字母, 同样的字符大写与小写属于不同的两种含义。



(9) 注释:为了便于阅读和理解程序,可以在程序的任何地方插入注释,以说明程序、函数、程序段、语句的功能或采用的算法。C语言的注释是用“/*”和“*/”括住的任意字符列。程序编译时,注释不参与编译,也不会出现在目标程序中,所以程序中的注释不影响编译后的目标程序,因此也不影响程序的功能。

例 1-1 中用到了两个系统函数:scanf()函数和 printf()函数。有关这两个函数的具体语法将在第二章讨论,这里仅介绍它们的功能。

scanf()函数的作用是接收从键盘上输入的数据并保存于指定变量中,这是从外界向程序输入数据的方法之一;printf()则是向标准输出设备上输出信息。

一般说来,向程序输入数据的方法有三种:

1. 赋值

用赋值语句为程序中的变量提供数据。这种方法运行简单,但是在数据较多且数据无规律时,程序书写很复杂;而且要求数据在设计程序时必须已知,否则没法编程;当数据变化时,修改了程序中的赋值语句,程序必须重新编译后再运行,修改才有效。所以这种方法适合于数据较少且数据改动不多的情况下使用。

例如: `int x, y;`

`x=10;`

`y=20;`

可以将程序中的变量 x 和 y 分别赋值为 10 和 20。

2. 键盘输入

C语言提供了若干从键盘上输入数据的方法,如 scanf()函数就是一种方法。这种方法不是在程序中给出数据,而是在程序运行时才从输入设备(键盘)上向程序中提供数据,因此在编程时数据尚未知时适合用这种方法;一旦用这种方法写好了程序后,从键盘输入不同的数据,就可以对不同的数据进行处理,而不需要重新编译程序。这种方法常用于类似“对任意数据”、“对若干数据”之类的问题进行输入的情况。

例如: `int x, y;`

`scanf ("%d %d", &x, &y );`

可以从键盘上输入两个任意整数并保存于变量 x 和 y 中。

3. 文件输入

从文件中读取数据,这种方法将在学习文件操作时进行介绍。

1.4 C语言实验环境——Turbo C 操作界面

Turbo C 是常用的 C 语言集成开发环境,它集编辑、编译、连接、调试和运行于一体,同时操作简便易于掌握,因此是常用的 C 语言学习环境。

进入 Turbo C 安装文件夹,运行 TC.EXE 便可启动 Turbo C 开发环境。

Turbo C 窗口如图 1-1 所示。

1. 主操作界面

Turbo C 主操作界面由主菜单区、编辑窗口区、信息窗口区及热键区四部分组成。

Turbo C 的采用菜单式操作,主菜单包括 File (文件操作)、Edit (编辑操作)、Run (运行操作)、Compile (编译操作)、Project (项目管理)、Options (选项)以及 Debug (调试)等项。

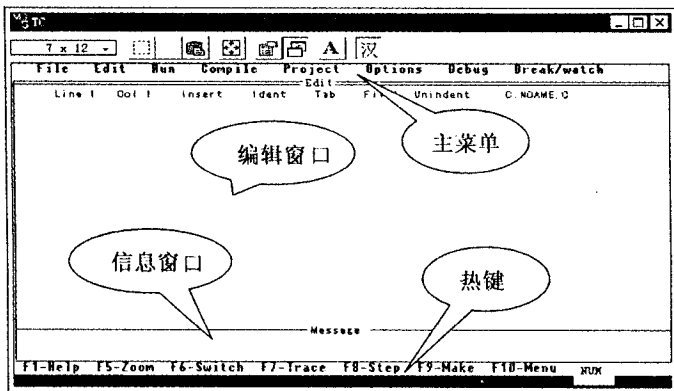


图 1-1 Turbo C 的界面

热键区提供常用操作的快速功能键方式,如 F1(帮助)、F2(保存文件)、F3(装入文件)、F5(缩放窗口)、F6(切换窗口)、Alt+F5(用户屏幕)、Alt+X(退出)等。

编辑窗口区是编辑程序的空间,编译和连接中的输出信息显示在信息窗口中。

2. 主窗口

当前操作窗口为主窗口(工作窗口),主窗口用双线框表示,按 F6 可切换主窗口,按 F5 可缩放主窗口,主窗口是编辑源程序的区域。

3. 菜单选择

方式一:按 F10 进入主菜单区,通过菜单项的大写字母选择相应菜单项,或通过光标控制键(→、←、↑、↓)移动高亮度光带至相应菜单项,按回车选择相应菜单项。

方式二:用热键 Alt 加相应菜单项的大写字母,选择相应菜单项。

对子菜单项的选择,操作方法类似。

4. 编辑

Turbo C 的编辑操作实现常用的文字编辑功能。

5. 编译

Turbo C 将编译、连接一道完成。按热键 F9 或选择 Compile 项进入编译。编译时发生错误,返回编辑过程,修改程序后,再编译程序。

6. 运行

按热键 Ctrl+F9 或选择 Run 项下的子项 Run,运行当前的程序。按 Alt+F5 键或选择 Run 项下的子项 User Screen 进入用户屏幕,查看程序运行结果,查看结束后按任意键返回编辑界面。

运行时发生错误,返回编辑步骤,修改程序后,再编译、运行程序。对于没编译的程序,选择运行会先自动进行编译操作,再运行程序。

7. 退出

上机过程完成后,按热键 Alt+X 或选择 File 项的子项 Quit 退出 Turbo C,返回操作系统状态。

1.5 算法描述工具简介

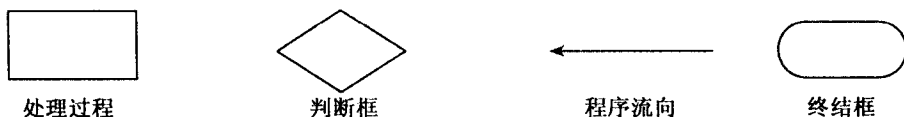
用计算机解题就是将程序员解题的步骤转变为计算机能理解的指令的过程,程序员对一个解题的过程有一个透彻的理解才能写出正确的程序。一般来讲,要写程序,必须先写算法。算法是对特定问题求解步骤的一种描述。只有给出了详尽有效正确的算法,才能转化为正确的程序,规范的算法描述也是程序员之间交流的有效途径。常用的算法描述工具有:程序流程



图、N-S图和伪代码。

1.5.1 程序流程图

程序流程图用以下四种基本图形来表示程序的处理过程：



处理过程指的是一个具有一定功能的程序单位。判断框表示一个逻辑判断。程序流向用来表示程序的执行次序。终结框表示一个程序的终结，如开始和结束。用程序流程图可以很清晰地表示程序的执行过程。例如判断一个数是否素数的算法用程序流程图描述如图 1-2 所示。

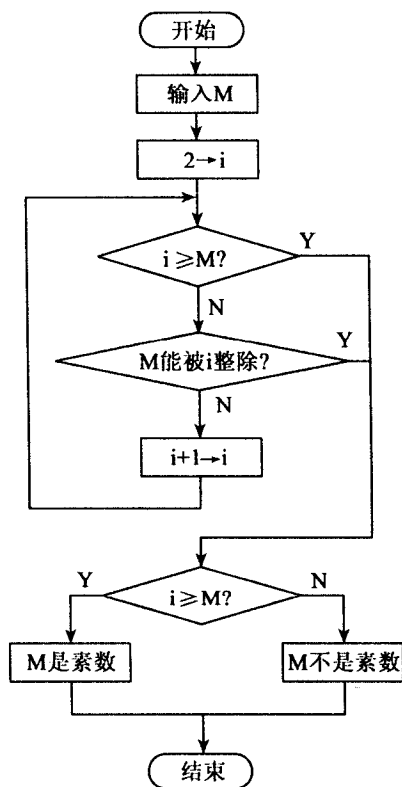


图 1-2 判断一个数是否为素数的流程图

程序流程图可以很方便地描述程序的执行过程，因而也很容易描述程序的功能，但从程序流程图过渡到程序还有一段距离，在程序中还不能清晰地反映用什么结构来实现程序。

结构化程序设计方法把程序的基本结构分为三种：顺序结构、选择结构（分支结构）和循环结构（重复结构），任何程序都可以由这三种结构经过嵌套和组合而成。能够更清晰地反映程序的三种结构的程序描述工具是 N-S 图和 PAD 图，用它们来描述程序不仅使程序结构更清晰，而且每种结构都可以用对应的语句来实现。