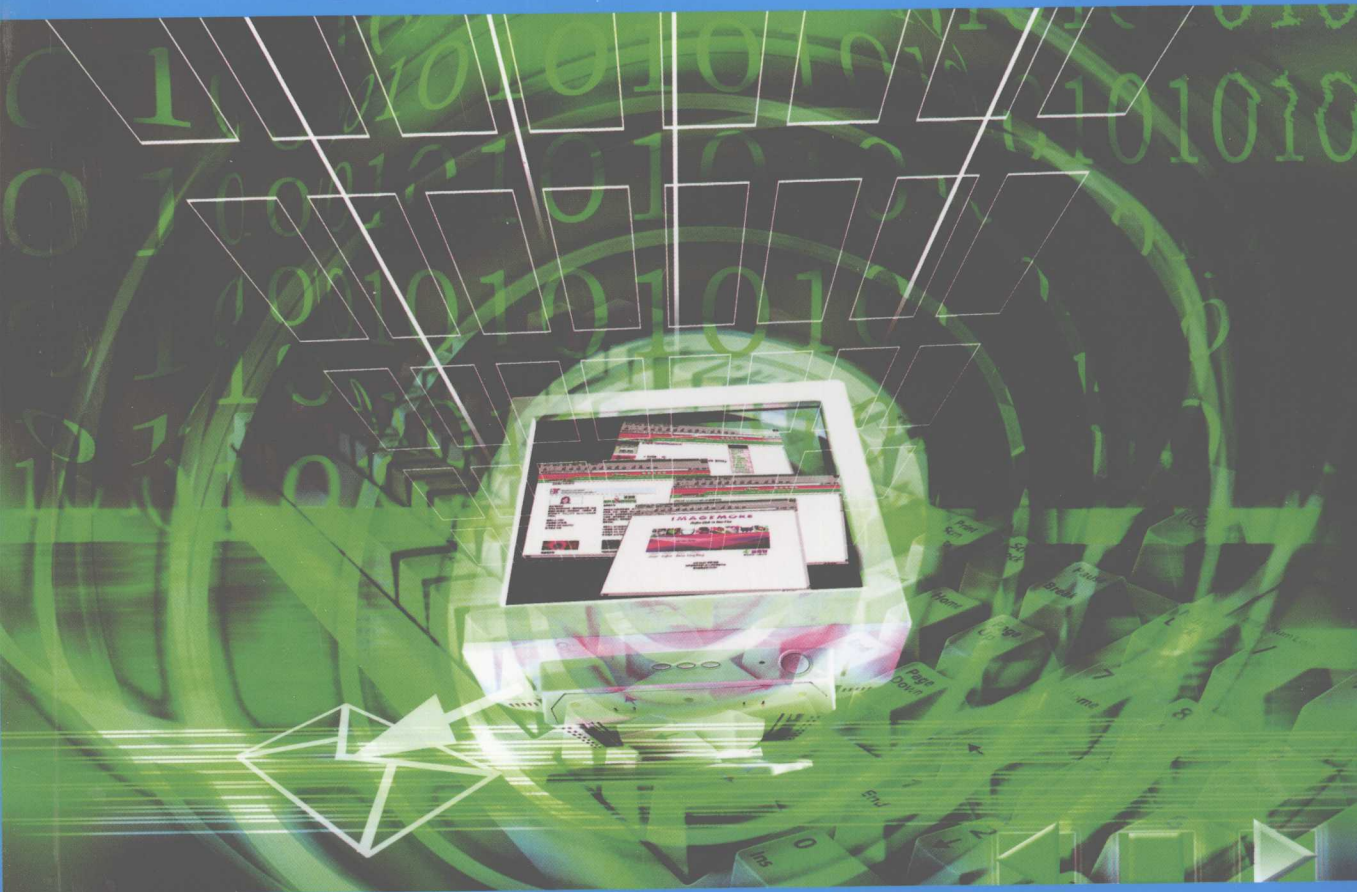




高职高专
计算机基础教育系列规划教材

C语言简明教程

新世纪高职高专教材编审委员会组编
主编 田湛君 郭晓利



大连理工大学出版社

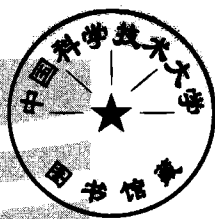
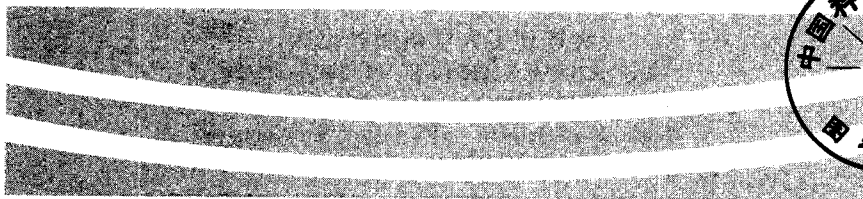


高职高专计算机基础教育系列规划教材

C 语言简明教程

新世纪高职高专教材编审委员会组编

主 编 田湛君 郭晓利 副主编 梁 霞 张 晖



C YUYAN JIANMING JIAOCHENG

大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

图书在版编目(CIP)数据

C 语言简明教程 / 田湛君, 郭晓利主编. —大连: 大连理工大学出版社, 2008. 4

高职高专计算机基础教育系列规划教材

ISBN 978-7-5611-4044-4

I. C… II. ①田… ②郭… III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 041162 号

大连理工大学出版社出版

地址: 大连市软件园路 80 号 邮政编码: 116023

电话: 0411-84708842 邮购: 0411-84703636 传真: 0411-84701466

E-mail: dutp@dutp.cn URL: <http://www.dutp.cn>

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸: 185mm×260mm	印张: 10.75	字数: 234 千字
附件: 光盘一张		印数: 1~4000
2008 年 4 月第 1 版		2008 年 4 月第 1 次印刷

责任编辑: 潘弘喆	马 双	责任校对: 李 飞
封面设计: 苏儒光		

ISBN 978-7-5611-4044-4

定 价: 19.80 元

我们已经进入了一个新的充满机遇与挑战的时代,我们已经跨入了 21 世纪的门槛。

20 世纪与 21 世纪之交的中国,高等教育体制正经历着一场缓慢而深刻的革命,我们正在对传统的普通高等教育的培养目标与社会发展的现实需要不相适应的现状作历史性的反思与变革的尝试。

20 世纪最后的几年里,高等职业教育的迅速崛起,是影响高等教育体制变革的一件大事。在短短的几年时间里,普通中专教育、普通高专教育全面转轨,以高等职业教育为主导的各种形式的培养应用型人才的教育发展到与普通高等教育等量齐观的地步,其来势之迅猛,发人深思。

无论是正在缓慢变革着的普通高等教育,还是迅速推进着的培养应用型人才的高职教育,都向我们提出了一个同样的严肃问题:中国的高等教育为谁服务,是为教育发展自身,还是为包括教育在内的大千社会?答案肯定而且惟一,那就是教育也置身其中的现实社会。

由此又引发出高等教育的目的问题。既然教育必须服务于社会,它就必须按照不同领域的社会需要来完成自己的教育过程。换言之,教育资源必须按照社会划分的各个专业(行业)领域(岗位群)的需要实施配置,这就是我们长期以来明乎其理而疏于力行的学以致用问题,这就是我们长期以来未能给予足够关注的教育目的问题。

如所周知,整个社会由其发展所需要的不同部门构成,包括公共管理部门如国家机构、基础建设部门如教育研究机构和各种实业部门如工业部门、商业部门,等等。每一个部门又可作更为具体的划分,直至同它所需要的各种专门人才相对应。教育如果不能按照实际需要完成各种专门人才培养的目标,就不能很好地完成社会分工所赋予它的使命,而教育作为社会分工的一种独立存在就应受到质疑(在市场经济条件下尤其如此)。可以断言,按照社会的各种不同需要培养各种直接有用人才,是教育体制变



革的终极目的。

随着教育体制变革的进一步深入,高等院校的设置是否会同社会对人才类型的不同需要一一对应,我们姑且不论。但高等教育走应用型人才培养的道路和走研究型(也是一种特殊应用)人才培养的道路,学生们根据自己的偏好各取所需,始终是一个理性运行的社会状态下高等教育正常发展的途径。

高等职业教育的崛起,既是高等教育体制变革的结果,也是高等教育体制变革的一个阶段性表征。它的进一步发展,必将极大地推进中国教育体制变革的进程。作为一种应用型人才培养的教育,它从专科层次起步,进而应用本科教育、应用硕士教育、应用博士教育……当应用型人才培养的渠道贯通之时,也许就是我们迎接中国教育体制变革的成功之日。从这一意义上说,高等职业教育的崛起,正是在为必然会取得最后成功的教育体制变革奠基。

高等职业教育还刚刚开始自己发展道路的探索过程,它要全面达到应用型人才培养的正常理性发展状态,直至可以和现存的(同时也正处在变革分化过程中的)研究型人才培养的教育并驾齐驱,还需要假以时日;还需要政府教育主管部门的大力推进,需要人才需求市场的进一步完善发育,尤其需要高职教学单位及其直接相关部门肯于做长期的坚忍不拔的努力。新世纪高职高专教材编审委员会就是由全国 100 余所高职高专院校和出版单位组成的旨在以推动高职高专教材建设来推进高等职业教育这一变革过程的联盟共同体。

在宏观层面上,这个联盟始终会以推动高职高专教材的特色建设为己任,始终会从高职高专教学单位实际教学需要出发,以其对高职教育发展的前瞻性的总体把握,以其纵览全国高职高专教材市场需求的广阔视野,以其创新的理念与创新的运作模式,通过不断深化的教材建设过程,总结高职高专教学成果,探索高职高专教材建设规律。

在微观层面上,我们将充分依托众多高职高专院校联盟的互补优势和丰裕的人才资源优势,从每一个专业领域、每一种教材入手,突破传统的片面追求理论体系严整性的意识限制,努力凸现高职教育职业能力培养的本质特征,在不断构建特色教材建设体系的过程中,逐步形成自己的品牌优势。

新世纪高职高专教材编审委员会在推进高职高专教材建设事业的过程中,始终得到了各级教育主管部门以及各相关院校相关部门的热忱支持和积极参与,对此我们谨致深深谢意,也希望一切关注、参与高职教育发展的同道朋友,在共同推动高职教育发展、进而推动高等教育体制变革的进程中,和我们携手并肩,共同担负起这一具有开拓性挑战意义的历史重任。

新世纪高职高专教材编审委员会

2001 年 8 月 18 日

前 言

C 语言是国内外广泛使用的一种计算机语言。C 语言功能丰富,表达能力强,使用灵活方便,应用面广,目标程序效率高,可移植性好。既具有高级语言的特点,又具有低级语言的特点;既可以编写系统软件,也可以编写应用软件。因此,学好 C 语言已成为广大学生的迫切需要。

本教材根据作者多年的教学实践经验,针对学生在学习 C 语言过程中遇到的实际困难而编写,以适应学生的接受能力和全面提高学生的综合素质为目的,精选要点,把握重点,克服难点,压缩冗点。全书结构清晰,层次分明,通俗易懂,每章末尾还配有大量精选的习题和技能训练。通过学习本教材,读者可以快速掌握 C 语言的基础知识,学会 C 语言的编程技术,提高解决实际问题的能力。

本教材共分为 12 章,分别是:概述;数据类型、运算符与表达式;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;函数;编译预处理;指针;结构体与链表;位运算;文件。

本书可以作为大学本科、高职高专学生的教材,也可以作为全国计算机等级考试的参考教材,同时也可作为自学者学习 C 语言的参考书。

由于编者水平有限,书中疏漏在所难免,敬请读者批评指正。

编 者

2008 年 4 月



第1章 概 述	1
1.1 C语言概述	1
1.1.1 C语言发展史	1
1.1.2 C语言程序的组成	2
1.1.3 C语言程序的特点	2
1.2 C语言的基本符号与关键字	3
1.2.1 字符集	3
1.2.2 关键字	3
1.2.3 用户标识符	3
1.2.4 ASCII字符集	4
1.3 算法及其表示	4
1.3.1 算法的概念和特征	4
1.3.2 算法的组成要素	4
1.3.3 算法的表示	5
1.4 C语言程序的开发与运行	6
第2章 数据类型、运算符与表达式	9
2.1 C语言的数据类型	9
2.2 常量与变量	9
2.2.1 常量	9
2.2.2 变量	10
2.3 整型数据	10
2.3.1 整型常量	10
2.3.2 整型变量	11
2.3.3 整型变量的使用	11
2.4 实型数据	12
2.4.1 实型常量	12
2.4.2 实型变量	13
2.5 字符型数据	13
2.5.1 字符常量	13
2.5.2 字符变量	14

2.5.3 字符串常量	15
2.6 算术运算符与算术表达式	15
2.6.1 C 语言运算符简介	16
2.6.2 算术运算符和算术表达式	16
2.7 赋值运算符和赋值表达式	19
2.7.1 赋值运算符	19
2.7.2 赋值表达式	19
2.8 逗号运算符和逗号表达式	20
2.8.1 逗号运算符	20
2.8.2 逗号表达式	20
第3章 顺序结构程序设计	22
3.1 顺序结构的语句	22
3.2 字符数据的输入输出	23
3.2.1 putchar()函数	23
3.2.2 getchar()函数	23
3.3 格式输入与输出	24
3.3.1 printf()函数	24
3.3.2 scanf()函数	26
3.4 顺序结构程序设计	29
第4章 选择结构程序设计	34
4.1 问题的引出	34
4.2 关系运算符和关系表达式	34
4.2.1 关系运算符	34
4.2.2 关系表达式	35
4.3 逻辑运算符和逻辑表达式	35
4.3.1 逻辑运算符	35
4.3.2 逻辑表达式	36
4.4 if 语句	37
4.4.1 if 语句	37
4.4.2 if 语句的嵌套	40
4.4.3 条件运算符	41
4.5 switch 语句	42
4.6 选择结构程序设计举例	43
第5章 循环结构程序设计	48
5.1 goto 语句	48
5.2 while 语句	49
5.3 do...while 语句	50
5.4 for 语句	51

5.5 循环的嵌套	53
5.6 break 语句和 continue 语句	54
5.6.1 break 语句	54
5.6.2 continue 语句	55
5.7 循环程序举例	55
第 6 章 数 组	63
6.1 一维数组	63
6.1.1 一维数组的定义	63
6.1.2 一维数组中元素的引用	63
6.1.3 一维数组元素的初始化	64
6.1.4 一维数组的应用	64
6.2 二维数组	66
6.2.1 二维数组的定义	67
6.2.2 二维数组元素的引用	67
6.2.3 二维数组的初始化	67
6.2.4 二维数组应用	67
6.3 字符数组与字符串	71
6.3.1 字符数组的定义	71
6.3.2 字符数组的初始化	71
6.3.3 字符数组的输入输出	72
6.3.4 字符串处理函数	72
第 7 章 函 数	80
7.1 函数的定义与调用	80
7.1.1 函数的定义	80
7.1.2 函数的返回值与函数类型	81
7.1.3 对被调用函数的说明和函数原型	81
7.1.4 函数的调用	82
7.1.5 函数的形参与实参	82
7.2 函数的嵌套调用和递归调用	83
7.2.1 函数的嵌套调用	83
7.2.2 函数的递归调用	84
7.3 数组作为函数参数	85
7.3.1 数组元素作为函数参数	85
7.3.2 数组名作为函数的形参和实参	86
7.4 局部变量与全局变量	86
7.4.1 局部变量	86
7.4.2 全局变量	87
7.5 变量的动态存储与静态存储简介	88

7.5.1 静态变量	88
7.5.2 自动变量	88
7.5.3 寄存器变量	89
7.5.4 外部变量	89
第8章 编译预处理	93
8.1 宏定义与符号常量	93
8.1.1 无参宏定义	93
8.1.2 有参宏定义	94
8.2 文件包含	95
8.2.1 文件包含处理命令的格式	95
8.2.2 文件包含的优点	95
8.3 条件编译	96
8.3.1 #ifdef 语句	96
8.3.2 #ifndef 语句	96
8.3.3 #if 语句	96
第9章 指 针	100
9.1 指针和指针变量的概念	100
9.1.1 内存地址——内存中存储单元的编号	100
9.1.2 变量地址——系统分配给变量的内存单元的起始地址	100
9.1.3 变量值的存取——通过变量在内存中的地址进行	101
9.1.4 指针运算符	101
9.2 指针变量的定义与应用	102
9.2.1 指针变量的定义	102
9.2.2 指针变量作为函数参数	103
9.3 数组的指针和指向数组的指针变量	104
9.3.1 概念	104
9.3.2 一维数组元素的引用	105
9.3.3 对指向数组的指针变量进行算术运算和关系运算	106
9.3.4 数组作为函数参数	106
9.3.5 二维数组的指针及其指针变量	107
9.3.6 二维数组指针作为函数参数	109
9.4 字符串的指针和指向字符串的指针变量	109
9.5 返回指针值的函数	110
9.6 指针数组与主函数 main() 的形参	112
9.6.1 指针数组	112
9.6.2 主函数 main() 的形参	113
9.6.3 指向指针的指针变量	113
9.7 函数的指针和指向函数的指针变量简介	114

9.8 指针小结	114
第 10 章 结构体与链表	120
10.1 结构体类型及其变量	120
10.1.1 结构体类型与结构体变量的定义	120
10.1.2 结构体变量的引用与初始化	122
10.2 结构体数组	123
10.3 指向结构体类型数据的指针	124
10.3.1 指向结构体变量的指针	124
10.3.2 指向结构体数组的指针	124
10.4 链表处理	125
10.4.1 链表结构	125
10.4.2 对链表的基本操作	126
10.5 共用型和枚举型	129
10.5.1 共用型	129
10.5.2 枚举型	129
10.6 定义已有类型的别名	130
第 11 章 位运算	134
11.1 数值在计算机中的表示	134
11.1.1 位与字节	134
11.1.2 数值的表示	134
11.2 位运算	135
11.2.1 位运算符和位运算	135
11.2.2 应用举例	137
第 12 章 文 件	141
12.1 C 语言文件概述	141
12.1.1 文件的概念	141
12.1.2 ANSI C 的缓冲文件系统	142
12.2 文件的打开与关闭	142
12.2.1 文件的打开(fopen()函数)	143
12.2.2 文件的关闭(fclose()函数)	144
12.3 文件的读写操作	144
12.3.1 读写字符	144
12.3.2 读写字符串	146
12.3.3 读写数据块	148
12.3.4 对文件进行格式化读写	148
12.4 位置指针与文件定位	149
12.4.1 位置指针复位(rewind()函数)	149
12.4.2 随机读写(fseek()函数)	149

12.4.3 返回文件当前位置(ftell()函数)	150
12.5 出错检测	150
12.5.1 ferror()函数	150
12.5.2 clearerr()函数	150
附录 1 常用字符与 ASCII 代码对照表	156
附录 2 运算符的优先级及其结合性	157
附录 3 常用的 C 库函数	158
参考文献	160

本章要点

- C 语言程序结构和特点
- C 语言程序的基本符号与关键字
- C 语言程序的编辑及运行

● 学习方法建议

学习本章内容应该从简单 C 语言程序入手,重点掌握 C 语言程序的特点,C 语言程序中的基本符号与关键字,以及编辑及运行 C 语言程序的简单方法。

1.1 C 语言概述

C 语言是当今世界上最为流行的面向过程的程序设计语言之一。它功能强大、可读性好、可移植性强,具有高级语言的所有优点,同时又具有低级语言的功能,如可以直接处理字符、位运算、地址和指针运算等。在结构上具有模块化、结构化的特征,既可以用来编写应用软件,又可以用来编写系统软件。C 语言是一种成功的描述语言,也是一种实用的程序设计语言。

1.1.1 C 语言发展史

在计算机技术的发展过程中,开发性能卓越的操作系统是技术人员不断追求的目标。在众多的操作系统中,UNIX 是成功中的典范。为描述和实现 UNIX 操作系统的需要,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,于 1970 年开发了 B 语言,并用 B 语言编写了 UNIX 操作系统。1972 年至 1973 年,贝尔实验室的 M. D. Ritchie 在 B 语言基础上设计出 C 语言,1973 年,他和 K. Thompson 合作用 C 语言将 UNIX 操作系统 90% 以上的源代码重新改写。UNIX 操作系统的一些主要特点,如可读性强、易于修改、具有良好的可移植性等,在一定程度上得益于 C 语言,所以 UNIX 操作系统的成功与 C 语言是密不可分的。

最初的 C 语言附属于 UNIX 的操作系统环境,而它的产生却可以更好地描述 UNIX 操作系统。现在的 C 语言已独立于 UNIX 操作系统,成为微型、小型、中型、大型和超大型(巨型)计算机上通用的一种程序设计语言。M. D. Ritchie 和 K. Thompson 也因他们在 C 语言和 UNIX 系统方面的卓越贡献获得了很高的荣誉。1982 年,他们获得了《美国电子学杂志》颁发的成就奖,成为该奖自颁发以来首次因软件工程成就而获奖的人。1983 年,他们又获得了计算机界的最高荣誉奖——图灵奖。

随着计算机应用领域的不断扩大,作为人与计算机进行信息交流工具之一的 C 语言

同样得到了迅速的发展。从最初的只是为描述和实现 UNIX 操作系统而设计的一种程序设计语言,到后来成为风靡全球的面向过程的计算机程序设计语言,C 语言取得了巨大的成功,成为世界上应用最广泛的几种计算机语言之一。

1.1.2 C 语言程序的组成

下面通过几个简单的例子来说明 C 语言的组成。

例 1.1 最简单的 C 语言程序:在计算机屏幕上输出“Hello!”。

```
main()
{
    printf("Hello! \n");    /* 输出 Hello! */
}
```

例 1.1 是一个最简单的 C 语言源程序。其中 main()是主函数,main 是函数的名称。用花括号“{}”括起来的内容是函数体,函数体应由若干条语句组成,这是计算机要执行的部分,每条语句以分号“;”结束。“/*...*/”之间的内容是语句的注释部分,供阅读程序之用。计算机并不执行注释部分的内容,编程时,为了使程序易读,应养成加注释的习惯。

例 1.2 用自定义函数求两个数中的最大值并输出。

```
main()
{
    int a,b,c;
    a=2;b=6;
    c=max(a,b);    /* 调用自定义函数 max() */
    printf("max=%d\n",c);
}

max(int x,int y)    /* 对 max()函数的定义 */
{
    int z;
    if(x>y) z=x;
    else z=y;
    return(z);
}
```

运行结果为:

max=6

同例 1.1 相比,例 1.2 除含有 main()函数外,增加了函数 max(int x,int y),整个程序由主函数 main()和函数 max(int x,int y)构成。需要说明的是,main()函数是程序执行的入口点,无论程序包含多少个函数,程序都从 main()函数开始执行。

1.1.3 C 语言程序的特点

从以上例子可以看出,C 语言程序的结构特点为:

(1) C 语言程序是由函数构成的,一个源程序可以包含若干个函数,但必须有且只有一个函数为主函数 `main()`,一个程序总是从主函数开始执行。

(2) 函数可以自定义,也可以调用 C 语言已有的库函数等。

(3) 函数一般都有函数体。函数体用花括号“{}”包含,从左花括号“{”开始,到右花括号“}”结束;函数体中有定义(或说明)和执行两大部分语句。

(4) 每一语句以分号“;”结束。

(5) 书写格式自由,可以在一行的任意位置书写;一行可以写一个语句,也可以写多个语句。因此,要注意养成良好的书写习惯,使程序便于阅读。

(6) 注释用“`/* ... */`”表示,它们之间的内容为注释部分,编译时系统对注释部分不作处理。

1.2 C 语言的基本符号与关键字

1.2.1 字符集

字符集是构成 C 语言的基本元素。用 C 语言编写程序时,除字符型数据外,其他所有成分必须由字符集中的字符构成。C 语言的字符集由下列字符构成:

(1) 英文字母: `A~Z, a~z`

(2) 数字字符: `0~9`

(3) 特殊符号:

空格	!	%	*	&	~	_(下划线)
+	=	-	~	<	>	/ \
,	"	;	.	,	()	[] {}

1.2.2 关键字

关键字是 C 语言已经定义的、具有特殊功能和含义的单词、单词缩写或者单词组合。以下列出的是 C 语言的关键字:

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

1.2.3 用户标识符

用户标识符即用户根据需要自己定义的变量名、常量名、函数名、数组名等。C 语言的用户标识符必须按以下规则命名:

- (1)必须以英文字母或下划线开始,并由字母、数字或下划线组成。例如:chABC, intX, a1 等都是合法的标识符,而 5Str, -chabc, +intJ 等则是非法的标识符。
- (2)每个标识符可以由多个字符组成,但只有前 8 个字符为有效标识符。
- (3)大写字母和小写字母代表不同的标识符,如 abc 和 ABC 是两个不同的标识符。
- (4)不能使用 C 语言的关键字作为用户标识符。

1.2.4 ASCII 字符集

在计算机中,所有的信息都用二进制代码表示。二进制编码的方式较多,应用最为广泛的是 ASCII 码。我们使用的字符在计算机中就是以 ASCII 码方式存储的。

ASCII 码是美国标准信息交换码(American Standard Code for Information Interchange)。它已被国际标准化组织(ISO)认定为国际标准,详见附录 1。

1.3 算法及其表示

1.3.1 算法的概念和特征

算法是指为了解决某个特定问题而采用的确定且有效的步骤。计算机算法可分为两大类:数值运算和非数值运算。数值运算的目的是求数值解,例如求方程的根、求圆的面积、求 n 的阶乘等。非数值运算包括的范围十分广泛,主要用于事务管理,例如人事管理、图书管理、学籍管理等。

算法有以下五个特性:

- (1)有穷性。一个算法应当包含有限个操作步骤;也就是说,在执行若干个操作之后,算法将结束,而且每一步都在合理的时间内完成。
- (2)确定性。算法中的每一条指令必须有确切的含义,不能有二义性,对于相同的输入必须能得出相同的结果。
- (3)有效性。算法中的每一步都应当有效执行,并得到确定的结果。例如,若 $b=0$,则 a/b 是不能有效执行的。
- (4)有零个或多个输入。对于计算机实现算法所需要的处理数据,有些程序在执行时需要通过输入得到,而有些程序不需要输入数据。
- (5)有一个或多个输出。算法的目的是求解(结果),结果要通过输出得到。

1.3.2 算法的组成要素

算法含有以下两大要素:

- (1)操作:每个操作的确定不仅取决于问题的需求,还取决于它取自哪个操作集,它与使用的工具系统有关。计算机算法要由计算机实现,组成它的操作集是计算机所能进行的操作。在高级语言中所描述的操作主要包括各种运算,如算术运算、关系运算、逻辑运算、函数运算、位运算和 I/O 操作等。计算机算法是由这些操作组成的。

(2)控制结构:每一个算法都是由一系列的操作组成的。同一操作序列,不同的执行顺序,就会得出不同的结果。控制结构即如何控制组成算法的各操作执行的顺序。在结构化程序设计中,一个程序只能由三种基本控制结构组成。这三种基本控制结构可以组成任何结构的算法,解决任何问题。

三种基本控制结构如下:

(1)顺序结构:顺序结构中的语句是按书写的顺序执行的,即语句执行顺序与书写顺序一致。这是一种最简单的结构,不能处理复杂问题。

(2)选择结构:最基本的选择结构是当程序执行到某一语句时,要进行判断,从两种路径中选择一条。计算机的判断能力就是通过选择结构实现的。

(3)循环结构:这种结构是将一条或多条语句重复地执行若干次。这种结构充分利用了计算机速度快的优势,将复杂问题用循环结构来实现。

1.3.3 算法的表示

算法可以用各种描述方法进行描述,常用的有自然语言、伪代码、传统流程图和 N-S 流程图等。传统流程图由以下几种基本框图组成,如图 1-1 所示。



图 1-1 传统流程图的基本符号

例 1.3 求 $5!$ 的算法(要求用传统流程图表示),如图 1-2 所示。

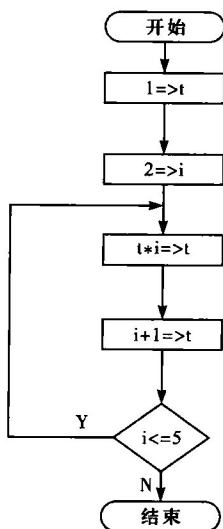


图 1-2 传统流程图

传统流程图用流程线指明各个框的执行顺序。对流程线没有严格的限制,因此使用