

C YU YAN JIAN MING JIAO CHENG



语言简明教程

● 蒋国瑞 主编



河北科学技术出版社

前 言

C语言是系统描述和通用程序设计语言,是当今在国内外得到迅速推广使用且最具影响的一种计算机语言。它具有先进的控制流、优良的数据结构、丰富的数据类型、实用多样的运算符和表达式。C语言功能强大、语言简练、使用灵活、应用面广。它既有高级语言的优点,又有低级语言的特点。由于它的通用性和可移植性好,已被广泛地用于系统程序设计和应用程序的开发。

随着计算机知识的广泛普及以及计算机技术的深入发展,计算机爱好者彻底、迅速地掌握一门计算机语言,已经成为当务之急。无论置身于哪个专业领域的计算机爱好者,C语言应该成为您首选的计算机语言。

本书是为计算机爱好者和大中专院校的学生编写的,其特点是:

1. 通俗易懂、深入浅出。本书取材讲究、重点突出,适合学员心理和接受能力;内容全面、详略适中,努力避开枯燥无味的陈述,增加趣味程序的实例。

2. 语言简练,充分体现“精讲多练”的原则。以大量新颖的实例说明,使读者体会和认识C语言的基本概念、语法规则、程序结构、使用方法和应用技巧。

3. 注重操作、讲究实用。本书取材十分注意使读者能够尽快地掌握C语言的知识,并学会使用这些知识来解决实际问题。为此我们从第一章开始,不断列举解决实际问题的短小精悍的完整程序,以提高读者分析问题、解决问题和编写程序的能力。

4. 书中所列程序(在Turbo C 2.0上运行已经通过)界面友好,提示信息齐全。有些程序的运行结果未在书中给出,先由读者通过阅读、分析程序,然后再上机操作进行验证,以便逐渐提高阅读程序和分析程序的能力,并养成良好的编程风格。

5. 考虑到程序的完整性和趣味性,本书在前面某些章节中,有个别程序涉及到尚未介绍的细节,读者应该把注意力集中在基本知识的学习上,不要停留在尚未介绍的细节上。

6. 本书第一章至第九章,包含了一般教材所涉及的全部基本知识。开设C语言课程,前九章是必学的内容(章末某些程序举例可以选学)。第十章介绍了C语言系统的文本窗口及图形功能,对于初学者这一章可以作为选学的内容。实际上这章内容在编写实用程序中是十分有用的。在第十章之后,本书给出了六个附录。附录1是趣味程序,它包含了八皇后问题、绘图等17个有趣的程序,学习并操作这些程序,不仅可以激发读者学习C语言的兴趣,而且可以更进一步使读者认识、了解乃至更好地使用C语言。附录2介绍Turbo C 2.0的基本操作,在上机前应该首先学习它。附录5给出了Turbo C 2.0编译错误信息,在调试程序时可以经常参考它。附录6指出了Turbo C 2.0与Turbo C++的主要区

别,说明了在 Turbo C 2.0 系统上运行的程序,怎样略加修改便可在 Turbo C++ 系统上运行。

本书由蒋国瑞主编、张新禄和张有会副主编,参加编写的还有孟林燕、刘宏忠、张宗镛、王道平、聂五军。本书是在作者《C 语言》教学讲稿基础上,经过大家多次整理、试用、修改而成的。河北师范大学张新禄同志整理并多次试用初稿、调试了全部程序,编写了第十章及附录。河北师范大学张有会同志对初稿进行了认真地修改和整理。石家庄铁道学院孟林燕同志、河北师范大学刘宏忠同志提供了很好的程序。河北科技大学张宗镛同志、河北经贸大学王道平同志、河北师范大学聂五军同志在本书的编写、整理、修改过程中也做了一定的工作。本书的完成是河北四所高校上述同志共同努力的结果。

尽管我们希望努力编写一本好教材,然而由于时间仓促,在内容、结构、写作体例、程序设计等诸方面,可能还存在一些问题,敬请专家和读者指正。

蒋国瑞

1997 年 1 月

目 录

第一章 概 述

1.1 C 语言由来	(1)
1.2 C 语言的特点	(1)
1.3 C 语言源程序的输入、编译和运行	(2)
1.3.1 C 程序的基本形式	(2)
1.3.2 建立第一个 C 程序	(3)
1.3.3 列磁盘文件目录	(4)
1.3.4 修改 C 程序	(4)
习题一	(6)

第二章 基本数据、运算符和表达式

2.1 标识符	(7)
2.2 C 语言的基本数据类型	(8)
2.3 常量与变量	(8)
2.3.1 常量与变量的概念	(8)
2.3.2 常量的类型	(9)
2.3.3 变量的类型	(11)
2.4 输出输入函数	(13)
2.4.1 printf 输出函数	(13)
2.4.2 scanf 输入函数	(16)
2.5 运算符与表达式	(19)
2.5.1 算术运算符与算术表达式	(19)
2.5.2 强制类型转换运算符	(20)
2.5.3 赋值运算符与赋值表达式	(21)
2.5.4 关系和逻辑运算符	(22)
2.5.5 逗号运算符与逗号表达式	(23)

2.5.6 条件运算符和条件表达式	(24)
习题二	(24)

第三章 程序控制语句

3.1 C 语句的概念	(27)
3.2 C 程序设计的三种基本结构	(28)
3.3 条件语句	(29)
3.3.1 if 语句	(30)
3.3.2 switch 语句	(33)
3.4 循环语句	(35)
3.4.1 while 语句(当型循环结构的语句)	(35)
3.4.2 do~while()语句(直到型循环结构的语句)	(36)
3.4.3 for 语句	(38)
3.5 无条件转移语句	(41)
3.5.1 goto 语句	(41)
3.5.2 break 语句	(41)
3.5.3 continue 语句	(42)
3.5.4 return 语句	(43)
3.6 程序举例	(44)
习题三	(45)

第四章 函数与变量

4.1 概述	(47)
4.2 函数定义	(48)
4.2.1 函数类型说明	(49)
4.2.2 函数参数说明	(51)
4.2.3 函数体与函数的返回值	(52)
4.2.4 函数原型	(53)
4.3 变量的分类	(54)
4.3.1 局部变量与全局变量	(54)
4.3.2 变量的存储	(56)
4.3.3 const 与 volatile 修饰符	(58)
4.4 函数调用	(58)
4.4.1 函数调用方法	(59)
4.4.2 函数的嵌套调用	(60)
4.4.3 函数的递归调用	(61)

4.5	内部函数和外部函数	(62)
4.5.1	内部函数	(63)
4.5.2	外部函数	(63)
4.6	部分常用库函数	(64)
	习题四	(70)

第五章 数 组

5.1	一维数组	(72)
5.1.1	一维数组的定义	(72)
5.1.2	一维数组元素的引用	(72)
5.1.3	一维数组的初始化	(74)
5.2	二维数组	(75)
5.2.1	二维数组的定义和引用	(75)
5.2.2	二维数组的初始化	(76)
5.2.3	二维数组程序举例	(76)
5.3	字符数组	(78)
5.3.1	字符数组的定义	(78)
5.3.2	字符串和字符串的结束标志	(79)
5.3.3	字符串函数 I	(81)
5.4	数组作为函数参数	(83)
5.4.1	数组元素做函数实参	(83)
5.4.2	用数组名作函数参数	(83)
5.4.3	多维数组作函数参数	(85)
5.5	程序举例	(86)
	习题五	(88)

第六章 指 针

6.1	指针与指针变量的定义及引用	(90)
6.1.1	指针、变量的指针的概念	(90)
6.1.2	指针变量的定义	(91)
6.1.3	指针变量的引用	(92)
6.1.4	指针变量作函数参数	(94)
6.2	指针变量的运算	(95)
6.2.1	指针变量的赋值	(95)
6.2.2	指针变量表达式	(95)
6.2.3	指针变量的算术运算	(95)

6.3	数组的指针与指向数组的指针变量	(97)
6.3.1	数组的指针	(97)
6.3.2	指向数组的指针变量	(97)
6.3.3	数组的指针作函数参数	(99)
6.3.4	指向多维数组的指针变量	(100)
6.4	指向字符串的指针变量	(103)
6.5	指向函数的指针变量和返回指针值的函数	(106)
6.6	指针数组和指向指针的指针	(108)
6.7	与指针相关的部分库函数	(109)
6.7.1	字符串函数 II	(109)
6.7.2	内存分配库函数	(111)
6.8	main()主函数的参数	(113)
	习题六	(114)

第七章 构造数据类型

7.1	结构体类型	(117)
7.1.1	结构体类型的定义	(117)
7.1.2	结构体变量的定义	(118)
7.1.3	结构体变量的引用	(119)
7.1.4	结构体变量的初始化	(120)
7.1.5	结构体数组	(121)
7.1.6	指向结构体类型的指针变量	(122)
7.1.7	指针与链表	(125)
7.2	共用体类型	(128)
7.2.1	共用体类型变量的定义和引用	(128)
7.2.2	指向共用体类型的指针变量	(130)
7.3	枚举类型	(131)
7.4	位运算	(133)
7.5	用 typedef 定义类型	(136)
7.6	程序举例	(137)
	习题七	(143)

第八章 编译预处理

8.1	宏定义	(144)
8.1.1	不带参数的宏定义	(144)
8.1.2	带参数的宏定义	(146)

8.2 文件包含	(150)
8.3 条件编译	(152)
8.3.1 #if、#else、#elif 和 #endif	(153)
8.3.2 #ifdef 和 #ifndef	(153)
习题八	(155)

第九章 文件操作

9.1 文件的打开和关闭	(158)
9.1.1 文件指针	(158)
9.1.2 文件的打开	(159)
9.1.3 文件的关闭	(160)
9.1.4 stdin、stdout、stderr 和 stderr 文件指针	(161)
9.2 文件的读写	(161)
9.2.1 fgetc()和 fputc()函数	(161)
9.2.2 getc、putc 和 getchar、putchar 宏	(163)
9.2.3 fgets()和 fputs()函数	(164)
9.2.4 fread()和 fwrite()函数	(165)
9.2.5 fprintf()和 fscanf()函数	(166)
9.2.6 getw()和 putw()函数	(167)
9.3 文件的定位	(168)
9.3.1 rewind()函数	(169)
9.3.2 ftell()函数	(169)
9.3.3 fseek()函数	(169)
9.4 文件的检测	(171)
习题九	(173)

第十章 C 系统的文本窗口及图形功能

10.1 文本模式下的程序设计	(174)
10.1.1 窗口及模式控制函数	(174)
10.1.2 属性控制	(175)
10.1.3 文本输出与管理	(177)
10.1.4 状态查询	(179)
10.2 图形模式下的程序设计	(180)
10.2.1 如何使用 Turbo C 图形函数库	(180)
10.2.2 图形系统控制	(181)
10.2.3 绘图与填充	(184)

10.2.4 管理屏幕及视口	(190)
10.2.5 图形模式下的正文输出	(193)
10.2.6 颜色控制函数	(195)
10.2.7 图形模式下的错误处理	(198)
10.2.8 图形模式下的状态查询	(199)
习题十	(200)
附录 1 趣味程序举例	(201)
附录 2 Turbo C 2.0 基本操作	(216)
2.1 Turbo C 的建立	(216)
2.2 Turbo C 的菜单系统及其使用	(216)
2.3 菜单命令和编辑命令	(218)
2.3.1 Turbo c 的菜单命令	(218)
2.3.2 Turbo c 的编辑命令	(222)
2.4 Turbo C 2.0 菜单结构一览表	(插页)
附录 3 C 语言中的关键字	(224)
附录 4 C 语言运算符一览表	(225)
附录 5 Turbo C 编译错误信息	(226)
5.1 致命错误	(226)
5.2 一般错误	(226)
5.3 警告	(228)
附录 6 Turbo C++ 与 Turbo C 2.0 的主要区别	(230)
参考文献	(231)

第一章 概 述

随着计算机技术的迅速发展和计算机应用的广泛普及,具有系统描述和通用程序设计功能的C语言越来越得到重视。当前绝大多数计算机上都配备了C语言。学习C语言、掌握C语言、使用C语言是形势对各专业领域的计算机爱好者的要求。

作为C语言简明教程的开始,本章简单介绍C语言产生和发展的背景、C语言的特点和用途、C语言程序的输入、编译和运行。

1.1 C语言由来

1972年,D. M. Ritchie在美国贝尔(Bell)实验室里,在《Basic Combined Programming Language》(简称B语言)的基础上,设计出C语言,它保持了B语言的优点,克服了B语言过于简单,数据无类型等缺点。最初的C语言只是为描述和实现UNIX(AT&T公司开发研制的)操作系统提供一种语言而设计的。在贝尔实验室里经过多次改进,直到1975年UNIX第六版公布后,C语言的突出优点才得到人们的普遍重视。

1977年,出现了不依赖于具体机器的C语言编译文本《可移植C语言编译程序》,大大简化了C语言移植到其它机器上的工作。

1978年,C语言先后移植到大中小微型计算机上。

同年B. W. Kernighan & D. M. Ritchie(合称K&R)以UNIX第五版中的C语言编译程序为基础,合著了影响深远的《The C Programming Language》一书,对C程序语言做了详尽的描述。随着微机的日益普及,大量的C语言工具得以问世,在源程序水平上,这些工具程序几乎可以说是奇迹般地高度兼容。然而,由于没有统一的标准,这些工具之间也有不一致的地方,为了改进这种情况,1983年美国国家标准协会ANSI(American National Standards Institute)根据C语言问世以来各种版本对C语言的发展和扩充,制定了ANSI C。

1988年,K&R按照ANSI C修改了他们的经典著作《The C Programming Language》。人们称ANSI C为新标准,称1978年写的《C语言》中所介绍的为旧标准。C语言从此风靡全球,成为计算机程序设计人员最喜欢和使用最广泛的语言。

1.2 C语言的特点

C语言的主要特点:

(1)C语言是中级语言。C语言既有高级语言(如Ada、Modula-2、Pascal、COBOL、FORTRAN、BASIC)面向问题、程序结构化、模块化的特点,也有低级语言(汇编语言)接近硬件的特点。它生成的代码具有很好的可移植性,是适用于作系统程序设计的基本语言。

(2)C语言简捷、紧凑,使用方便、灵活。下面是C语言与Pascal语言的比较:

C 语言	Pascal 语言	含 义
{ }	begin end	复合语句
int a[20];	var a:array [1..20] of integer;	定义 a 为含 20 个元素的整型数组

(3)C 语言运算符丰富,共 34 种,包括了除控制与输入输出操作以外的几乎所有的操作,如括号、赋值、强制类型转换等。

(4)C 语言的数据类型丰富,具有计算机语言的各种数据结构。

(5)C 语言的重要结构成分是函数——子程序,C 语言的函数作为程序结构的基本单位,使得用 C 语言编程非常容易实现程序的模块化。

(6)C 语言另一结构化的方法是复合语句,即一系列语句置于{ }中,它使许多运算过程清晰、简洁和高效。

(7)C 语言具有编译预处理功能。这是 C 语言的一个重要特点。

(8)C 语言是编程者的语言。C 语言提供给编程者所要的:很少限制、很少缺陷、模块结构,最受专业编程者的欢迎。

汇编语言,调试困难,是非结构程序,可读性差、难以维修和移植。C 语言克服了汇编语言的缺点。它的显著特点是:功能丰富,描述能力强;编译容易,结构模块化;移植方便,编程效率高;既有高级语言的优点,又有低级语言的功能。

C 语言适用于描述系统程序,包括操作系统、语言处理、系统实用程序、应用软件,还可用于数据处理、科学工程数值计算等很多领域。

1.3 C 语言源程序的输入、编译和运行

1.3.1 C 程序的基本形式

C 程序的基本形式是:

```
main( )
{
    变量说明语句;
    执行语句;
}
```

其中 main() 是主函数,一个程序必须有一个主函数。执行语句可以包含其他函数。

例 1.1 下面是一个求 $3877 \div 79$ 的商和余数的 C 程序。

```
main( )
{
    /* 程序的开始 */
    int a,b,c,d;          /* 变量说明 */
    a=3877;
    b=79;
    c=a/b;                /* 把商的值赋给 c */
    d=a%b;                /* 把余数赋给 d */
    printf(" %d/%d=%d...%d \n",a,b,c,d);
}
```

```
} /* 程序的结束 */
```

程序中后五句是执行语句,printf()是一个函数,实现输出功能。用/* */括起来的部分为C语言的注释部分,它可以提高程序的阅读和理解能力。

1.3.2 建立第一个C程序

Turbo C语言是按照ANSI C新标准设计的一种C语言,本书中的程序均在Turbo C 2.0环境下编译、运行过。

1. 建立第一个Turbo C程序

操作步骤如下:

(1)将系统置于TC子目录的提示符C:\TC>。

(2)键入TC并回车,稍时屏幕显示Turbo C的主屏幕。

(3)在主屏幕下按F10及F或者直接按Alt-F,出现File子菜单后选择Load项,或者直接按功能键F3均可出现一窗口,输入文件名(如MYPROG1.C)后回车,则屏幕提示为新文件,并进入编辑状态。

(4)通过键盘输入程序。

例 1.2 输入程序:

```
main( )
{
printf("Now let's learning the C—Language. \n");
printf("It is my first C program. \n");
}
```

2. 程序存盘

为防止意外事故丢失程序,最好将输入的程序存贮在磁盘上。在编辑窗口下,可直接按Turbo C的热键F2或F10.F.S(先按F10键,再按F键,最后按S键),也可以按Alt-F.S键,存盘时屏幕最底行提示:

"Saving edit file"

3. 编译程序

对源程序进行编译有几种方法:①在编辑状态下直接按Alt-F9或②按F10返回主菜单,选择Compile项或者③按Alt-C,然后在Compile的下拉菜单中选择Compile to. OBJ项后回车,这时屏幕会出现一个编译窗口,几秒钟后,窗口中显示一闪烁信息:

"Success: Press any key"

表示编译成功。此时按任意键,编译窗口消失,光标返回。

如果编译时,信息窗口产生警告Warning或出现Error,则表明源程序有错误,按F6,光标返回文件出错行(处),并处在编辑状态,修改后再重新编译。

用/* */括起来的注释部分不进行编译。

4. 运行程序

源程序经编译无误后,可以投入运行。具体操作有三种方法:

(1)在编辑状态下直接按Ctrl-F9。

(2)按F10返回主菜单,选择Run项。

(3)按 Alt-R,然后在 Run 的下拉菜单中选择 Run 项后回车。

程序投入运行,屏幕会出现一个连接窗口,显示 Turbo C 正在连接的程序所需的库函数。

连接完毕后,Turbo C 即运行程序,结束后即返回执行前的状态,按 ALT-F5 可观察运行结果,这时屏幕显示:

Now let's learning the C—Language.

It is my first C program.

按任意键返回。如果想退出 Turbo C,可按 ALT-X。

注:经编译成功的程序在运行时还有可能出现错误,这时应根据提示的信息找出原因,再按上面介绍的方法进一步修改。

1.3.3 列磁盘文件目录

在 C 程序的编辑、编译、连接过程中,将产生一些不同扩展名的新文件,要想了解这些文件的初步信息,可通过列磁盘文件目录的方式进行。列磁盘文件目录的操作如下:

(1)按 F10 键,激活主菜单,按 F、Q 键并回车,或直接按 ALT-X,执行 File 菜单的 Quit 命令退出 Turbo C 返回 DOS 状态,也可以不退出 Turbo C 而选择 File/Os Shell 项暂时挂起正在执行的工作,而回到 DOS 状态下(执行 DOS 操作结束后使用 EXIT 命令返回挂起前的状态)。

(2)如要列出文件名为 MYPROG1 的所有文件的有关信息,可键入 DIR MYPROG1.

* 按回车键,则屏幕显示:

MYPROG1.C (此文件为 C 语言源文本文件,在 DOS 提示符下可用 TYPE 命令显示)

MYPROG1.OBJ (此文件是编译后产生的二进制机器指令目标代码,在 DOS 提示符下不可以用 TYPE 命令显示)

MYPROG1.EXE (此文件是 Turbo C 连接产生的实际可执行文件,在 DOS 提示符下键入 MYPROG1 并按回车键则可执行此程序)

1.3.4 修改 C 程序

例 1.3 修改程序 MYPROG1.C,使它运行产生下面的结果,并改文件名为 MYPROG2.C。

```
* * * * *
Frankly, my dear...
Turbo C 2.0 is a better C.
* * * * *
```

在编辑状态下,打开 MYPROG1.C,将程序修改如下:

```
main( )
{printf(" * * * * * \n");
printf("    Frankly, my dear... \n");
printf("    Turbo C 2.0 is a better C. \n");
```

• 4 •

```
printf(" * * * * * \n");
printf("\n");
}
```

然后做为一个新程序存盘,这时不能简单的按 F2 键以 MYPROG1.C 名存盘,这样会覆盖原 MYPROG1.C 文件,应起一个新的文件名,例如 MYPROG2.C,操作如下:

首先按 F10 键,进入主菜单,再按 F、W 键选取 File/Write to 或直接按 Alt-F 键选 Write to 命令执行,Turbo C 则提示用户输入新文件名,此时输入 MYPROG2.C 后,按回车键,则屏幕底行显示"Saving edit file"存盘,同时编辑器中的文件名改为 MYPROG2.C,运行 MYPROG2.C 可得到欲想的输出结果。

例 1.4 编一个程序,运行时先提示输入三个数。输入后,它输出三数中的最大者。

```
main( )                                /* 主函数,返回值为空类型 */
{ float a,b,c,max;                     /* 对变量 a、b、c、max 作类型说明 */
  printf("Please input three numbers a,b,c :");
  scanf ("%f%f%f",&a,&b,&c); /* 输入 a、b、c 的值,括号中的 & 是取址运算符 */
  max=a;
  if(max<b) max=b;
  if(max<c) max=c;
  printf("The maximum is %f \n",max);
}
```

在 C 语言程序中还可以包含汉字信息。例如使用 UC DOS,先启动 UC DOS 汉字操作系统,然后再运行 TC.EXE,则在 Turbo C 程序中可包含汉字信息。如:

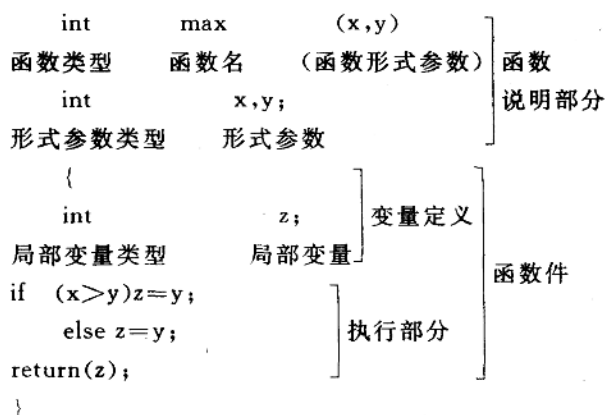
例 1.5 编一个给定两个整数,求出较大者的程序。

```
int max( )
int x,int y;                           /* 定义 max 函数,函数值为整型,x、y 为形参,并
                                         对形参变量 x、y 作类型说明 */
{ int z;                               /* 对 max 函数中用到的变量 z 作类型说明 */
  if (x>y)z=x;
  else z=y;=return(z);                 /* 将 z 的值返回到函数的调用处 */
}

main( )                                /* 主函数,返回值为空类型 */
{ int a,b,c;                           /* 对变量 a、b、c 作类型说明 */
  printf("请输入两个整数分别给变量 a 和 b: ");
  scanf ("%d %d",&a,&b);               /* 输入 a、b 的值,括号中的 & 是取地址运算符
  */
  c=max(a,b);                          /* 调用 max 函数,并将所得的值赋给 c */
  printf("a 与 b 中较大的整数为 %d \n",c); /* 输出 c 的值 */
}
```

由例 1.5 可知,在 C 语言中,函数是完成某种特定功能的子程序。C 程序是由函数组成的,函数是 C 程序的基本单位,这一特点使得 C 程序非常容易实现程序的模块化。

一个函数由两部分组成:函数的说明部分与函数体,以 max()函数为例分析如下:



习 题 一

- 1.1 请写出 C 语言的主要特点。
- 1.2 C 语言以函数为程序的基本单位,有什么好处?
- 1.3 上机运行本章例题,熟悉 Turbo C 环境下的上机方法和步骤。
- 1.4 在例 1.3、例 1.4 和例 1.5 中故意键入一点错误(例如:去掉";"、圆括号以及双引号或大括号的配对等)看程序执行的结果如何。
- 1.5 编写通过键盘输入两个整数并输出其和的程序。
- 1.6 编写通过键盘输入三个互不相同的整数,输出其中较大的整数并要求用汉字提示信息的程序。

1.7 运行下面程序,观察显示结果:

```
main() /* Main program starts here */
{
printf("Good form");
printf("can aid in ");
printf("understanding a program. \n");
printf("And bad form ");
printf("can make a program ");
printf("unreadable. \n");
}
```

第二章 基本数据、运算符和表达式

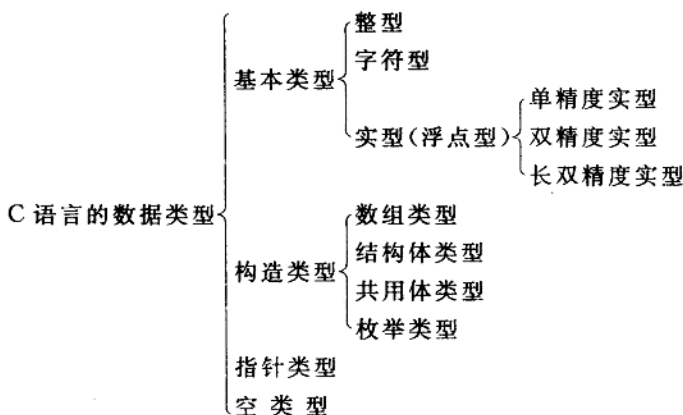
著名计算机科学家沃思(Nikiklaus Wirth)对程序设计提出一个公式:

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

随着计算机科学的发展,有的专家又指出:

$$\text{程序} = \text{算法} + \text{数据结构} + \text{程序设计方法} + \text{语言工具和环境}$$

由此可见,数据结构在程序设计中的重要性。C语言提供的数据结构是以数据类型出现的,C语言的数据类型如下:



2.1 标识符

C语言中用来标识函数名、常量名、变量名、数组名、文件名以及用户定义对象的名称等有效字符序列称为标识符。关于标识符有如下规定:

(1)它的第一个字符必须是字母或下划线,后续字符可以用字母、数字和下划线。如abc、_name、Sex、month、_a9611、student_1是正确的。而3class、K&R、#301、hi!th是不正确的。

(2)C语言中大小写字母是有区别的,如max、Max、MAX是三个不同的标识符。

(3)标识符的有效长度随系统而定,一般最长不超过8个字符,如果打了多于8个的字符做标识符,则系统一般只取前8个字符,多余字符忽略。Turbo C允许最长32个字符(这是Turbo C菜单选择结构O/C/S标识符约定的长度,可重设)做标识符。

(4)C语言标识符不能与保留关键字同名(C语言关键字见附录Ⅲ)。

2.2 C语言的基本数据类型

C语言的基本数据类型有整型、实型和字符型,各类型的分类见表2-1。

表 2-1 C语言基本数据类型分类

类 型		表示方法	占位	值 域
整 型	基本整型	int	16	$-2^{15} \sim (2^{15}-1)$
	短整型	short	16	$-2^{15} \sim (2^{15}-1)$
	长整型	long	32	$-2^{31} \sim (2^{31}-1)$
	无符号整型	unsigned	16	$0 \sim (2^{16}-1)$
	无符号短整型	unsigned short	16	$0 \sim (2^{16}-1)$
	无符号长整型	unsigned long	32	$0 \sim (2^{32}-1)$
实 型	单精度实型	float	32	$3.4E-38 \sim 3.4E+38$
	双精度实型	double	64	$1.7E-308 \sim 1.7E+308$
	长双精度实型	long double	80	$3.4E-4932 \sim 1.1E+4932$
字 符 型	基本字符型	char	8	$-128 \sim 127$
	无符号字符型	unsigned char	8	0.255

2.3 常量与变量

2.3.1 常量与变量的概念

先来看下面的程序:

例 2.1 某物品价格 300 元/件,下面是一个给定件数显示总价格的程序。

```
#define RATE 300
main()
{ int a, total;
  printf("Input number:");
  scanf("%d",&a);
  total=a * RATE;
  printf("Total price=%d\n",total);
}
```

其中 #define 是宏定义,它定义 RATE 代表常量 300。在运行时程序中凡是 RATE 都用 300 替换。在程序运行过程中,其值不变的量称为常量。常量一般用大写字母表示,例如 RATE。

例 2.2 将小写字母 a 转换成大写字母 A 的程序。

```
main()
```