

C语言与数据结构

金以文 编著

浙江大学出版社



C 语言与数据结构

金以文 编著

浙江大学出版社

内容简介

本书分为十七章,全面介绍了C语言的基本概念、程序设计方法以及常用的数据结构。本书是为非计算机专业的研究生和本科高年级学生以及从事CAD和CAM专业的工程技术人员为掌握现代语言、提高计算机应用水平而编写的教材。作者根据多年教学经验编写的这本书具有概念清晰、例题丰富、逻辑性强、深入浅出、通俗易懂的特点,也可供具有一定计算机基础的读者自学。

C语言与数据结构

金以文 编著

责任编辑 杨春亭

*

浙江大学出版社出版

浙江大学出版社计算机中心电脑排版

浙江省煤田地质勘探公司制图印刷厂印刷

浙江省新华书店发行

*

787×1092 16开 17.25印张 441千字

1994年11月第1版1996年10月第2次印刷

印数:1501—4500

ISBN 7-308-01377-4/TP·133 定价:17.50元

前 言

本书由两部分组成：C 语言程序设计和数据结构。

本书首先介绍了 C 语言。C 语言是一种通用程序设计语言，著名的 UNIX 操作系统就是用 C 语言编写的。由于它具有丰富的数据类型，灵活方便的多种运算符，精炼的语句表达式，高效率的代码以及可移植性等优点，受到了世界上软件专家的赞赏。随着 UNIX 系统的日益普及，许多过去使用 PASCAL 和 FORTRAN 的用户越来越多地转向 C 语言。因此，在大学里普及 C 语言就十分必要了。从第一章到第八章详细介绍了数据类型、运算符、流程控制、函数、指针、结构和输入输出函数等。为帮助读者掌握和熟悉程序设计概念，书中包含了大量程序实例。

本书第二部分介绍数据结构。《数据结构》是计算机软件的一门基础课程。本课程的目的是介绍一些常用的数据结构，阐明数据结构内在的逻辑关系，讨论它们在计算机中的存贮表示以及相应的各种运算。从第九章到第十七章分三部分介绍数据结构：第一部分是“线性结构”，介绍了顺序表、链表、堆栈和队列等概念以及线性分类和查找各种运算；第二部分是“非线性结构”，介绍了树、二叉树、穿线树以及树的周游、分类和查找算法；第三部分是“复杂结构”，介绍了图、网络、最短路径、关键路径以及拓扑排序等概念和算法。在介绍数据结构基本概念同时，给出了用 C 语言编写的各种数据结构的算法程序，读者在学习过程中读了大量的具有代表性的程序，能很快地熟悉概念，学会应用，提高程序设计能力。

本书层次清楚、深入浅出、通俗易懂、内容丰富，尤其对非计算机软件专业的研究生和本科高年级学生，从事 CAD 和 CAM 专业的工程技术人员是一本比较合适的教材和参考书，通过本书学习，对 C 语言和数据结构会有较全面的认识，为独立进行有关领域的软件设计打下良好的基础。讲授学时为 70 学时左右。

由于水平有限，书中不妥之处在所难免，热诚希望读者指正。

编 者

1994 年 10 月

目 录

第一章 C 语言简介

- 1.1 发展过程和特点 1
- 1.2 C 语言程序结构 2
- 1.3 C 语言基本符号 4
- 1.4 C 程序编辑、编译和执行 4

第二章 数据类型

- 2.1 基本数据类型 8
- 2.2 数组 10
- 2.3 类型转换 11

第三章 运算符和表达式

- 3.1 表达式 13
- 3.2 算术运算符 13
- 3.3 关系运算符和逻辑运算符 14
- 3.4 位逻辑运算符 15
- 3.5 自增和自减运算符 16
- 3.6 赋值算符和组合算符 17
- 3.7 其他运算符 18
- 3.8 优先级和结合性 19
- 3.9 程序举例 20

习题 21

第四章 流程控制

- 4.1 条件语句 24
- 4.2 循环语句 26
- 4.3 开关语句 30
- 4.4 转移语句 32
- 4.5 程序举例 35

习题 40

第五章 函数和程序结构

- 5.1 函数定义和调用 42
- 5.2 函数递归 46

5.3 变量的存储类别和作用域 48

5.4 分程序结构 54

5.5 C 预处理程序 55

5.6 程序举例 60

习题 66

第六章 指针

6.1 指针定义及其运算 68

6.2 指针和数组 71

6.3 指针数组 74

6.4 指针和函数 76

6.5 程序举例 79

习题 84

第七章 结构

7.1 结构和结构指针定义 87

7.2 结构数组 90

7.3 结构嵌套 93

7.4 字段 95

7.5 联合 96

7.6 枚举类型 98

7.7 类型定义 99

7.8 程序举例 100

习题 104

第八章 输入输出函数

8.1 标准输入输出 105

8.2 按格式输入输出 107

8.3 内存中格式转换 111

8.4 文件存取 111

8.5 低级输入输出 118

8.6 程序举例 122

习题 128

第九章 概论

- 9.1 什么是数据结构 130
- 9.2 数据结构的分类 133
- 9.3 数据结构与算法关系 133
- 9.4 数据结构发展概况 134

第一部分 线性结构 136

第十章 顺序表

- 10.1 向量 137
- 10.2 数组 140
- 10.3 栈 141
- 10.4 队列 148

习题 151

第十一章 链表

- 11.1 单链表 152
- 11.2 循环链表 158
- 11.3 双向循环链表 159
- 11.4 链式栈和链式队列 161

习题 164

第十二章 线性分类

- 12.1 分类概述 165
- 12.2 插入分类 166
- 12.3 交换分类 170
- 12.4 归并分类 173
- 12.5 选择分类 176

习题 177

第十三章 线性查找

- 13.1 顺序查找 178
- 13.2 二分法查找 180
- 13.3 分块查找 182
- 13.4 散列(HASH)查找 183

习题 192

第二部分 非线性结构 193

第十四章 树

- 14.1 树的基本术语 194
- 14.2 树的逻辑结构和物理表示 195
- 14.3 树结构的应用概述 197

习题 199

第十五章 二叉树

- 15.1 二叉树的概念 200
- 15.2 二叉树的物理表示 203
- 15.3 周游二叉树 204
- 15.4 穿线树 211
- 15.5 一般树的二叉树表示、周游和运算 215

习题 217

第十六章 树的分类和查找

- 16.1 分类二叉树 218
 - 16.2 哈夫曼树 224
 - 16.3 最佳分类二叉树 228
- 习题 231

第三部分 复杂结构 232

第十七章 图

- 17.1 图的基本术语 233
- 17.2 图的物理表示 235
- 17.3 图的周游和生成树 238
- 17.4 最短路径 244
- 17.5 拓扑排序 251
- 17.6 关键路径 255

习题 257

附录 C 库函数 259

参考文献 267

第一章 C 语言简介

1.1 发展过程和特点

C 语言是一种面向结构的程序设计语言,是一种功能很强而又比较简单的通用程序设计语言。它具有高级语言的许多特点,又具有比较低级的汇编语言的功能;既是成功的系统描述语言,又是一个实用的程序设计语言,这就是 C 语言具有的两重性。1973 年,D. M. Ritchie 用 C 语言改写了用汇编语言写的 UNIX 操作系统的 90%,成为有名的 UNIX 操作系统第五版本,并于 1974 年在 UCACM 上正式发表。当时普遍使用在美国各大学中。但是,不久它就被全世界广泛接受了,从近几年的发展形势来看,已经有甚多的 PASCAL 语言的爱好者,逐渐以 C 语言来替代。

C 语言的发展过程如下:

语言名	设计者	年份
CPL	C. Strachay 等	1968
BCPL	M. Richards	1969
B	K. Thompson	1970
C	D. M. Ritchie	1972

CPL(Combined Programming)是延续 ALGOL 语言而设计出来的一种语言,由于系统本身过于庞大且复杂,很难学习和编写,因而没有得到推广。在 CPL 的基础上发展了 BCPL(Basic Combined Programming Language)语言,后又演变为用于特殊用途的 B 语言。D. M. Ritchie 于 1972 年在 B 语言的基础上研制了 C 语言,并用 C 语言写成了第一个在 PDP-11 计算机上实现的 UNIX 操作系统。到 1978 年以后,C 语言被移植到小到 8 位微机,大至巨型机 Cray 1 等许多计算机上。目前,C 语言已独立于 UNIX 和 PDP-11,运行在各种操作系统上和各种机种上。特别是在 8 位至 32 位微机上使用非常普遍。

C 语言具有以下特点:

- (1) 具有丰富的数据类型,较多的运算符,较少的关键字,简洁的表达方式,便于编写各种功能程序;
- (2) 提供了良好的结构程序所需的基本流程控制语句;
- (3) 提供了指针,能做地址运算,能直接访问物理地址,使得它在直接操作硬件功能时,能实现同汇编语言相同的描述;
- (4) 程序结构是以函数为基本的集合体,函数能独立编译和调试,一个程序就可以由小到大,由简单到复杂堆积而成。函数间参量的传递是按“值”调用方式进行的,这使得被调用的函数不能改变调用函数程序的实际参数值;
- (5) 通过预处理可以进行宏调用,可防止书写错误,便于修改常数;

(6) 程序设计自由度大,表示方法简洁,使用方便,容易学习和掌握;

(7) 用 C 语言编程,程序员可以获得高效率的机器代码,其效率几乎接近于汇编语言代码;

(8) C 语言程序非常容易移植,基本上不作修改能用于各种型号的机器。

当然,C 语言也有缺点,语法关系的严格性不如 FORTRAN 和 PASCAL,错误检查不完善。

1.2 C 语言程序结构

用 C 语言编写的程序,称作 C 语言源程序,简称 C 程序。C 程序一般由一个或多个函数组成,这些函数可集中放在一个或分散在几个文件中。各文件都必须以 .c 结尾,如 sum.c。

首先写一个简单的程序:

```
/* add of a and b */
main()
{
    int a,b,sum;
    a=123;
    b=456;
    sum=a+b;
    printf("sum is %d\n",sum);
}
```

执行结果: sum is 579

这个程序的功能是把 a 和 b 两个数相加,然后输出和的结果。对上述程序说明如下:

(1) /* 与 */ 之间的内容是注释,注释行可以出现在程序的任何部分,注释界符 / 和 * 中间不能有空格,而 * 和注释之间要有空格。注释行不被编译,只起注解和帮助阅读的作用。

(2) main 是一特殊函数名,所有 C 语言程序都以 main 函数开始执行,其实质是 C 程序的首部(或主程序),main() 后面的圆括号()不能省去,参量放在括号内。该程序是一个无参量的函数。

(3) 花括号{}是函数体的界限符,用以包括构成函数的全部语句。

(4) 第 4 行是变量 a,b,sum 的说明语句,int 是 integal(整型)的缩写,说明 a,b,sum 是整型变量。

第 5 行是二个赋值语句,把整数 123 赋给变量 a,整数 456 赋给变量 b。

第 6 行是把 a 和 b 值相加,结果赋给变量 sum。

第 7 行输出变量 sum 的值。双引号"....."内所指出的字符串是控制格式,% 符后面的字母为格式符,这里 d 表示用十进制格式输出,\n 是 C 中换行字符。printf 是输出函数,是系统库提供的标准函数。

(5) 每个语句必须用分号(;)结尾。一行中可以有若干个语句。有一点特别注意,花括号"}"后面不能有分号。

在一个完整的 C 程序中,可以有若干个函数组成,但一定要有一个函数 `main()`,不管这个函数放在 C 程序中什么地方,C 程序总是从它开始执行。函数 `main()` 可以调用其他函数,其他函数之间也可以有调用和被调用的关系,函数本身也可以自调用,称为递归调用。

C 程序的完整结构为:

```
外部变量说明;
main( )
{
    内部变量说明;
    执行语句;
}
函数名(参量表)
参量说明;
{
    内部变量说明;
    执行语句;
}
```

为了以后各章举例方便,简单介绍一下输入和输出函数,详细介绍参见第八章。

输出函数调用形式:

`printf("转换格式控制串",变量名表);`

其中,`printf()`为输出库函数;双引号括起来的"转换格式控制串"有以下几种常用形式:

格式串	相应的参数输出形式
<code>%d</code>	十进制整数
<code>%c</code>	单个字符
<code>%s</code>	字符串
<code>%f</code>	十进制浮点数
<code>%o</code>	八进制数
<code>%x</code>	十六进制数

输入函数调用形式:

`scanf("转换格式控制串",变量名指针表);`

其中,`scanf()`为输入库函数;双引号括起来的"转换格式控制串"常用的格式与上述输出相同;接收输入的变量要用其指针型,即变量名前加"取地址运算符—&".例如:

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

用户从终端上打入的数(例如 123),按十进制整数解释,赋给变量 `x`。

1.3 C语言基本符号

基本符号有英文字母,阿拉伯数字以及特殊符号。其中有

(1) 英文字母 26 个(包括大写和小写);

(2) 阿拉伯数字 10 个,即从 0 到 9;

(3) 特殊符号主要是指运算符,如:

初等量表达式操作符: (), [], . , ->;

一元操作符: * , & , - , ! , ~ , ++ , -- , sizeof ();

二元操作符: * , / , % , + , - , > , < , <= , >= , == , != , & , ^ , | ,
&& , || , ? : ;

赋值组合操作符: = , += , -= , * = , / = , % = , >> = , << = , & = , ^ = ,
|= ;

逗号操作符: , ;

(4) 标识符

用来表示函数,类型和变量的名称。由字母数字表示,必须以字母开头(连字符_也称为字母),只有前 8 个字符有效。按一般习惯,变量名用小写字母,符号常数用大写字母。函数名和外部变量名不同机器有不同的限制(DEC PDP-11 前 7 个字符有效,可以用大、小写字母;Honeywell 6000 前 6 个字符有效,只用一种字母等)。

(5) 关键字标识符有:

- 描述数据类型定义的关键字: typedef;
- 描述变量存储方式的关键字: auto, extern, static, register;
- 描述数据类型关键字: char, short, int, long, unsigned, float, double, union, struct, enum;
- 描述语句的关键字: while, do, for, switch, break, continue, return, goto, if else, case;
- 其他: void, default 等。

用户使用的变量名和函数名不能用以上的关键字标识符。

1.4 C 程序编辑、编译和执行

通过文本编辑程序(edit 或 vi)将 C 程序作为源文件送入磁盘。编写程序时要注意格式,尽可能易于阅读和理解,以便于修改和维护。例如,

```
main( ) { int i,total=0;for(i=1;i<11;i++)total=total+1;  
printf("1+2+3+4+5+6+7+8+9+10=%d\n",total);}
```

编写时不停顿地书写或不必要地挤在一块,十分不清楚。应写成如下明白易懂的格式:

```
main( )  
{  
    int i,total=0;
```

```

for(i=1;i<11;i++)
    total=total+1;
printf("1+2+...+9+10=%d\n",total);
}

```

编程式样最好遵循以下规则：

- (1)按处理功能进行函数化；
- (2)一个函数的语句行数最好不超过 50 行左右；
- (3)不要在一行里写若干条语句(如果不影响理解,可以写 2—3 条语句)；
- (4)每个功能程序段利用空格键划分范围,把重复范围的开始行和结束行对齐。例如,下一个程序中 while 的第一个字母 w 与闭括号(})对齐,作为一个功能程序段。main()的开括号({)与闭括号(})对齐,作为一个主函数整体。

```

main( )
{
    int    a, b, x;
    scanf("%d %d", &a, &b);
    while(a! =0 && b! =0) {
        x=a+b;
        printf("x=%d \n", x);
        scanf("%d %d", &a, &b);
    }
}

```

输入源文件需要打入以下命令：

```
$ vi xxx.c
```

其中, \$ 为分时用户的提示符(可以用其他符号取代)；

vi(或 ed)为 UNIX 操作系统的文本编辑程序;表示调用 vi,编辑源文件 xxx.c;xxx 表示用户使用的源文件名。

文本编辑命令随不同的机器都稍有不同,可以参考所使用的机器手册。

通过文本编辑程序将 C 程序送入磁盘后,就可以进行编译了。C 语言的编译功能较强,可以经过编译直接产生目标文件,甚至能一次产生内存映象文件。下面介绍几种编译和执行的方法。

1. 在 UNIX 操作系统环境中使用的步骤

(1) 简单的方法

以 hello.c 为例,打入以下命令：

```
$ cc hello.c
```

直接产生内存映象文件 a.out(为可执行程序),最后,打入执行命令：

```
$ a.out
```

(2) 通用方法

可以在命令行加入选择项-o,要求按用户给出的文件名确定其内存映象文件名。编译命令为：

```
$ cc -o hello hello.c
```

编译后,产生的内存映象名为 hello,就不再是 a.out,执行命令为:

```
$ hello
```

有的机器上 UNIX 操作系统的改进文本,编译后产生的内映象文件即为源文件名,上例为 hello,这就不必使用 -o 选择项了。

2. 在 DOS 下用 MS C 4.0 编译程序运行 C 程序的步骤

(1)编辑 C 源程序。可以用行编辑程序 EDLIN 或 WORDSTAR。

(2)编译。要经过 4 道编译,或称 4 次扫描。假设源文件名为 f.c,则编译命令为:

```
cc1 f.c ✓ (第一次,作用为预处理)
cc2 f.c ✓ (第二次,语法分析)
cc3 f.c ✓ (第三次,编译)
cc4 f.c ✓ (第四次,优化编译)
```

如果在编译时出现出错信息,需重新编辑(修改)源程序。

(3)连接。将编译好的目标文件同库函数连接起来。最简单的链接命令为:

```
Link f,,cslib✓
```

cslib 是 C 系统库(c system library)的缩写。链接成功后得到的可执行文件名自动定为 f.exe (文件名与源文件名相同,后缀改为.exe)。

(4)执行程序。只需输入目标文件名:

```
f✓
```

3. 用 Turbo C 运行 C 程序的步骤

先将 Turbo C 装入磁盘。

(1)调用 Turbo C 程序,只需打入:

```
tc✓
```

屏幕顶部出现一排“命令”行菜单:

FILE EDIT RUN COMPIER PROJECT OPTIONS DEBUG

Load	F3
Pick	Alt-F3
New	
Save	F2
Write to	
Directory	,
Change dir	
Os shell	
Quit	Alt-x

用键盘上的“←”和“→”键来移动屏幕上的光标,光标指到哪一个命令字时,按回车键就表示执行该命令。开始时,光标指向“FILE”,表示对文件进行输入输出。

按回车键,屏幕上“FILE”下面出现一个窗口,见图 1.1 所示。它是一个子菜单,提供多项选择。用“↓”键将光标移到“Load”或“New”处,按回车键,表示要输入源程序。屏幕又出现一个小窗口(见图 1.2)。要你指定文件名。可输入:

```
f.c✓
```

如果原来不存在此文件名,则表示建立一个新文件。如果已存在此文件,则将此文件调入并显示在屏幕上。此时自动转为编辑(EDIT)状态。

图 1.1

(2)编辑源文件。根据需要输入或修改源程序。

Load File Name
 x. c

图 1.2

(3)编译源程序。按“F9”键即可进行编译、连接,并在屏幕上显示有无错误和有几个

错误。当按任何一个键后,屏幕显示源程序,光标停留在出错之处。在屏幕的下半部分显示出有错误的行和错误原因。根据此信息修改源程序。再按“F9”编译,如此反复进行到不出现错误为止。

(4)执行程序。按“F10”键,屏幕又出现命令行:

FILE EDIT RUN COMPLIER PROJECT OPTIONS DEBUG

用“→”键将光标移到“RUN”,按回车键,使执行已编译好的目标文件。此时,屏幕上会显示出程序应输出的运行结果。如果程序需要输入数据,则应在此时输入所需数据,然后接着执行程序,输出结果。

如果发现运行结果不对,要重新修改源程序,可以再按“F10”键,并用“←”使光标指到“EDIT”处,按回车键,即进入编辑状态,可以根据需要修改源程序,并重复上述 2、3、4 步,直到得到正确结果为止。

(5)可以用“Alt”和“X”键(同时按此二键),使脱离 Turbo C,回到操作命令状态,此时,可以用系统命令显示源程序和运行程序:

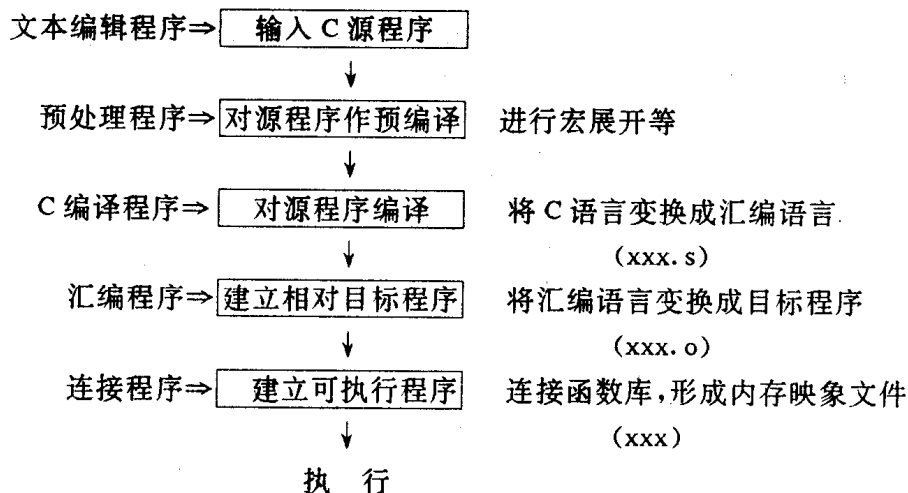
c>TYPE f. c (列出源程序清单)

c>f (执行目标程序 f. exe)

如果想再修改源程序,可以重新执行步骤 1,并输入源程序文件名即可。

以上步骤只需上机试一下即可明白。

从 C 语言的编译到执行,其过程如下:



第二章 数据类型

2.1 基本数据类型

C 语言中基本数据类型有四种:char(字符型),int(整数型),float(单精度浮点型)和 double(双精度浮点型)。

1. 字符型(char)

字符型数据有单个字符和字符串两种。

- 单个字符占用一个字节(8 位),通常用来表达字符集(ASCII 码或 EBCDIC 码字符集)中的一个字符,字符常量是用单引号括起来的单个字符,'a','b'等;也可以用来表达短整数,取值范围为-128 到+127。

- 字符串占用多个字节,要用字符型数组表达一个字符串,数组的每个元素(即为一个字节)存放串中一个字符,最后加入空白字符 NULL(\0)来标志字符串结束。字符串常量用双引号括起来,如"a b c","a"等。

注意区别'a'和"a",前者是单个字符 a,后者是 a 和\0 组成的字符串。

某些非图形字符可以用带反斜杠的扩展表示法,如下表所示。这些字符看上去好象是两个字符组成,但实际只起一个作用。它们在屏幕上显示不起来,一般起控制作用。注意区别'\0'和'0': '\0'表示具有编码值 0 的字符;而'0'表示的是字符 0(在 ASCII 码中编码为 48,在 EBCDIC 中为 240)。\ddd 为换码符,如'\014'表示 ASCII 码中的 FF。

非图形字符	扩展表示法	功 能
NL(LF)	\n	换 行
HT	\t	水平制表
BS	\b	退 格
CR	\r	回 车
FF	\f	走 纸
\	\\	反斜杠
'	\'	单引号
ddd	\ddd	三位 8 进制数

2. 整数型(int)

这是表达整数值的数据形式,它的长度是所使用计算机的机器字长,如字长为 16 位,取值范围为-32768 到+32767。

整常数通常用 10 进制表示,也可以用 8 进制和 16 进制表示。用 8 进制表示,开头要写上 0(零),其后再写上要表示的 8 进制数,8 进制数用数码 0~7 表示,在 C 语言中也可以用

8 和 9, 分别表示 8 进制 10 和 11。用 16 进制表示时, 开头要写上 0x(0X), 在 16 进制的表示法里, 除了数码 0~9 外, 还使用字母 A~F(或 a~f), 对应 10 进制的 10~15。

例如: 10 进制数: 31

8 进制数: 037

16 进制数: 0x1f 或 0X1f

在整型数据中, 根据数据的 2 进制位数多少有 short(短整数)、int(基本整数)、和 long(长整数)之分, 具体字长和具体的机器规定有关。

此外, 还有无符号(unsigned)整型数据类型, 也分为三类, 即: unsigned short, unsigned int 和 unsigned long。描述基本整型时, 可以省略 int, 此时, 取值范围为 0~65535。

3. 单精度浮点型(float)

这是表达实数值的数据形式, 由整数部分, 小数部分和指数部分组成, 占用二个字长(一般为 32 位), 小数点保留 7 位有效位, 取值范围为 -10^{38} 到 -10^{-38} ; 0 ; 10^{-38} 到 10^{38} 。

注意浮点数的表示方法, 合法的表示如:

1. 2e+3 (e 代替 10, 即 1.2×10^3);

1. 2e-3 (相当于 1.2×10^{-3});

12. 34 (缺省指数部分);

. 1234 (缺省整数和指数部分);

以下是不合法表示:

4. 5e (e 右边一定要有整数部分);

. e+3 (小数点两边至少有一个数字部分);

e-6 (整数部分和小数部分不能同时缺省);

0. 3e4. 5 (指数部分要整数型数值)。

4. 双精度浮点型(double)

是单精度浮点的二倍字长, 一般为 64 位, 10 进制的有效位为 15 位, 最大可表示到 $1e+308$ 。

实际上所有浮点计算都是以 double 进行的, 对于 float 变量先转换为 double 型(尾部添零), 然后运算, 当结果赋给 float 变量时, 对双精度结果进行截尾, 截尾时四舍五入。

下表给出了几个不同机器上的基本数据类型及其字长。

机 型	DEC PDP-11	Honeywell 6000	IBM 370	Interdata 8/32
类 型 \ 码	ASCII	ASCII	EBCDIC	ASCII
char	8 位	9 位	8 位	8 位
int	16	36	32	32
short	16	36	16	16
long	32	36	32	32
float	32	36	32	32
double	64	72	64	64

VAX-11 除了 int 是 32 位外, 其余与 PDP-11 相同。

在 C 程序中,所有变量在使用之前,必须加以类型说明,通常放在函数开始处,在可执行语句之前。变量说明语句的格式:

类型 变量名表;

举例如下:

```
char c;  
int lower, upper, step;  
float x, y;  
double z;  
unsigned int i, j;
```

其中, c 为字符型变量; lower, upper 和 step 为整型变量; x, y 为单精度浮点型变量; z 为双精度浮点型变量; i, j 为无符号整型变量。

2.2 数组

C 语言除了基本数据类型外,还可以由基本数据类型构造出复杂的数据形式,如数组就是其中一种。

数组由固定数量的同类型元素组成,即数组由若干个元素组成,且每一个元素具有相同的数据类型,并能用统一的数组名和下标唯一地确定数组中每一个元素。程序中使用数组时,首先要进行数组说明,编译时给所说明的数组分配一块连续存贮空间,一个元素存放一个数据。说明语句形式:

类型 数组名[元素个数];

如:

```
int a[5];  
char b[10];  
float c[10];
```

这三个说明语句,说明了数组 a、b、c 分别是整型、字符型和浮点型数组。其中 a、b、c 是数组名,方括号内的数字为元素个数,这些元素是从下标为 0 的元素开始,例如, a[5] 的 5 个元素分别是: a[0], a[1], a[2], a[3], a[4]。每个元素占用 2 个字节,数组 a 总共占用 10 个字节。字符型数组一般用来存放字符串,一个元素存放一个字符(即占用一个字节), b[10] 共占用 10 个字节。浮点型数组 c 的每个元素占用二倍机器字长(即 4 个字节), 10 个元素共占用 40 个字节。

C 语言提供了二维数组,说明语句为

类型 数组名[行下标][列下标];

如:

```
int    a[10][20];
char   matrix[10][40];
```

这二个说明语句表示:数组 a 为 10 行 20 列整型数组;数组 matrix 为 10 行 40 列字符型数组。数组 a 共有 200 个元素,它们是按行存放的,a[0][0],……a[0][19],a[1][0]……a[1][19],……a[9][0]……a[9][19];matrix 共有 400 个元素,两者占用同样的存储空间。

2.3 类型转换

在 C 程序里,字符型,整型和浮点型数据可以在同一个表达式中混合使用。当两个操作数类型不一致时,C 编译程序将按一定的规则自动进行类型转换。

转换规则可以归纳为以下原则:

把占用字节数较少的类型(称为较低类型)转换为占用字节数较多的类型(称为较高类型)。因此,有以下关系:

$$\text{char} < \text{int} < \text{unsigned} < \text{long} < \text{float} < \text{double}$$

左边为较低类型,右边为较高类型。当在一个表达式中具有不同类型的变量时,在运算前,首先把各个变量类型转换为表达式中出现的最高类型,当类型一致后,再进行运算。注意的是,当表达式中最高类型为单精度浮点型(float)时,为了提高运算精度,所有变量类型都转换为双精度浮点型(double),运算结果也为 double 型。

类型转换也发生在赋值语句的左、右边。转换规则是:把赋值符右边的类型转换为左边变量的类型。此时,要注意精度的损失。例如,程序段:

```
int    i;
char   c;
i=c;
```

其中 i 为整型变量,c 为字符变量。把 c 的值赋给变量 i,此时,精度不损失。

如果赋值语句为

```
c=i;
```

将整型变量 i 的低 8 位赋给字符变量 c,丢掉了高 8 位数。使用时,必须特别注意。

又如:

```
int    i;
float  x;
i=x;
```

把浮点变量 x 值赋给整型变量 i,只取 x 的整数部分,舍去了 x 的全部小数部分。

除了机器自动按规则进行转换外,也可以使用 cast 算符强使显式的类型按用户要求进