

Turbo

C

程序设计语言

杭必政 龚沛曾 杨志强 编著

表达能力强 目标质量高

同济大学出版社

Turbo C 程序设计语言

杭必政
龚沛曾 编著
杨志强

同济大学出版社

(沪) 204 号

内 容 简 介

Turbo C 语言是在 UNIX 操作系统的基础上发展起来的,它介于高级语言和低级语言之间,该语言的表达能力强,通用性和可移植性好,目标质量高,程序书写简洁、清晰,编译程序规模小,是近年来深受用户欢迎的一种语言。

全书共分 14 章。其内容有数据、表达式和运算符、语句和控制流、数组、函数、存储和数据通讯、编译预处理、指针变量、结构和联合、文件和输入输出、Turbo C 的屏幕与图形功能、Turbo C 2.0 环境,书中还附有丰富的 C 程序设计实例。

本书是高等院校非计算机专业的教材,也可作为从事计算机工作的技术人员参考。

责任编辑 冯时庆

封面设计 陈益平

Turbo C 程序设计语言

杭必政 龚沛曾 杨志强 编著

同济大学出版社出版

(上海四平路 1239 号)

新华书店上海发行所发行

上虞科技外文印刷厂印刷

开本: 787×1092 1/16 印张: 18 字数: 460 千字

1994 年 5 月第 1 版 1994 年 5 月第 1 次印刷

印数: 1—6000 定价: 11.00 元

ISBN 7-5608-1359-3/TP·133

前 言

C 程序设计语言(简称 C 语言)具有描述问题能力强、灵活性好、容易实现编译、目标质量高、通用性和可移植性好等优点。它不仅适用于编写系统软件,而且愈来愈广泛地被使用于编制各种应用程序。

本书较全面地介绍了 C 语言的主要内容,包括 Turbo C 的屏幕与图形功能及 TurboC 2.0 的环境。

全书以大量典型举例从易到难介绍了使用 Turbo C 的程序设计方法,本书不仅适用于非计算机专业大学生学习 C 语言的教材用书,也可作为其他计算机工作者的参考资料,也可作为有关人员的培训教材。

本书第 1 章到第 4 章和第 11 章由 龚沛曾编写;第 5 章到第 10 章和第 12 章由杭必政编写;第 13、14 两章由杨志强编写。

同济大学计算机系杨振山教授审阅了全稿,复旦大学夏宽理副教授校阅了全稿,他们都提出了不少宝贵意见。李宁东参加了部分章节的程序调试,也提出了一些合理的建议,在此一并表示谢意。由于时间紧迫,不妥之处在所难免,热忱欢迎读者提出批评和指正。

编者 1993 年 4 月

目 录

第1章 C语言概述	(1)
§1 C语言的由来	(1)
§2 C语言的特点	(2)
§3 C语言程序结构	(3)
第2章 C语言基本概念	(6)
§1 基本语法单位	(6)
§2 数据类型	(7)
§3 变量	(8)
§4 常量	(10)
§5 变量的初始化	(14)
§6 基本的I/O库函数	(14)
第3章 表达式和运算符	(22)
§1 表达式	(22)
§2 算术运算	(24)
§3 关系运算	(24)
§4 逻辑运算	(25)
§5 按位运算	(26)
§6 赋值运算	(32)
§7 条件运算	(33)
§8 自增、自减运算	(35)
§9 其他运算符	(36)
§10 不同类型的转换	(37)
§11 运算符综合举例	(40)
第4章 语句和控制流	(44)
§1 语句	(44)
§2 循环语句	(46)
§3 if语句	(54)
§4 switch语句	(58)
§5 其他辅助控制语句	(61)
§6 综合举例	(64)
第5章 数组	(70)
§1 数组的基本概念	(70)
§2 一维数组	(70)
§3 二维数组	(73)

§ 4 字符数组	(76)
第 6 章 函数	(82)
§ 1 函数的定义	(82)
§ 2 形式参数和实在参数	(85)
§ 3 递归函数	(87)
第 7 章 存储类别和数据通讯	(98)
§ 1 变量的生存期和作用域	(98)
§ 2 自动变量	(98)
§ 3 寄存器变量	(102)
§ 4 外部变量	(103)
§ 5 静态变量	(105)
§ 6 函数的存储类别	(106)
§ 7 局部变量和全局变量	(107)
第 8 章 编译预处理	(116)
§ 1 宏定义命令	(116)
§ 2 包含命令	(119)
§ 3 条件编译	(120)
第 9 章 指针变量	(126)
§ 1 指针的概念	(126)
§ 2 指针对象	(128)
§ 3 指针运算	(129)
§ 4 指针和函数	(132)
§ 5 指针和字符串	(135)
§ 6 指针和数组	(138)
§ 7 多级指针	(142)
§ 8 命令行参数	(144)
第 10 章 结构和联合	(156)
§ 1 结构类型的说明	(156)
§ 2 定义结构类型变量的方法	(157)
§ 3 结构变量的引用	(158)
§ 4 结构类型变量的初始化	(159)
§ 5 结构数组	(160)
§ 6 指向结构的指针	(161)
§ 7 指向结构数组的指针	(162)
§ 8 结构与函数	(163)
§ 9 结构应用(一)——链表	(169)
§ 10 结构应用(二)——二叉树	(183)
§ 11 结构应用(三)——位域	(190)
§ 12 联合	(192)

§ 13 枚举类型.....	(194)
§ 14 用 typedef 语句定义类型.....	(196)
第 11 章 文件和输入输出	(201)
§ 1 文件	(201)
§ 2 标准设备文件的 I/O 函数	(203)
§ 3 一般文件的输入输出	(208)
§ 4 文件的定位和修改	(217)
§ 5 低级输入输出	(220)
§ 6 综合举例	(222)
第 12 章 Turbo C 的屏幕与图形功能简介.....	(230)
§ 1 一个简单图形的画图程序	(230)
§ 2 Turbo C 的画图功能	(230)
§ 3 图形屏幕管理和视区设置	(233)
§ 4 Turbo C 的字符屏幕管理	(242)
第 13 章 C 程序设计举例	(252)
第 14 章 Turbo C 2.0 环境	(267)
§ 1 Turbo C 综述.....	(267)
§ 2 在不同配置的系统上建立 Turbo C 2.0	(271)
§ 3 Turbo C 2.0 集成开发环境.....	(272)
§ 4 Turbo C 2.0 命令行环境.....	(277)

第1章 C语言概述

§1 C语言的由来

随着计算机技术的迅速发展和广泛应用,高级程序设计语言的功能越来越强,但共同存在的问题是缺乏用于书写操作系统和编译程序等系统程序的工具,系统程序设计仍然主要依赖于汇编程序,使得程序的可读性和可移植性都比较差。在这种情况下,人们开始探索一种能用于系统程序设计,有足够表达能力而又高效率的程序设计语言,也即既具有一般高级语言的功能,又具有接近机器语言特性的语言。于是介于高级语言和低级语言之间的一种C程序设计语言(简称C语言)系统就迅速地发展起来。

C语言是美国 AT&T (American Telephone & Telegram) 贝尔 (Bell) 实验室的 D. M. Ritchie 在 UNIX 系统上研制成功的。C语言于 1972 年正式投入使用。1973 年, K. Thompson 和 D. M. Ritchie 为美国 DEC 公司的 PDP-11 计算机用 C 语言重写了 UNIX 操作系统。因此, C 语言既与 UNIX 系统有十分密切的关系,但又独立于 UNIX 系统。

图 1.1 表示了 C 语言的“家谱”。由图可见, ALGOL 是 C 语言的祖先。ALGOL 是一种面

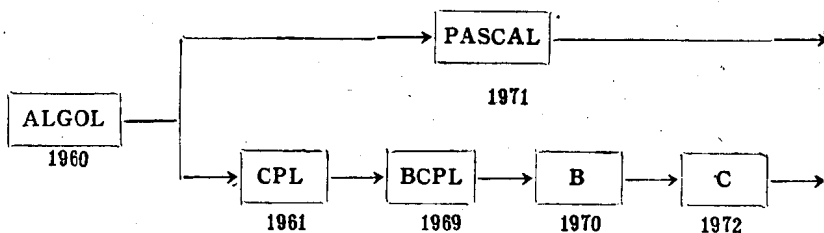


图 1-1

向问题的语言,与计算机硬件距离很远,不利于编写系统程序。CPL(Combined Programming Language)语言则企图改造 ALGOL,使它和计算机硬件更接近一些,但是它是一种比较大的语言,难于学习和实施。BCPL 语言对 CPL 语言作了精简,并保持了它的基本优点。1970 年,贝尔实验室的 Ken Thompson 在实施 UNIX 系统时对 BCPL 作了进一步简化,设计了非常简单又很接近于硬件的 B 语言,但 B 语言适用范围小。为此,1972 年 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言。C 语言吸收了 B 语言中合理而有效的部分,改变了 B 语言“无类型”的数据结构等不足,系统地引进了多种基本数据类型,并导出了其他组合类型和函数。

最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的,用 C 语言重写后的 UNIX 可读性好,移植性好,加之 UNIX 本身所具有的优点,使 UNIX 成为国际上使用最为广泛的操作系统,UNIX 的广泛流行,又进一步扩大了 C 语言的影响。C 语言已先后移植到不同机器上,各种 C 编译系统相继问世,它适应的机器可以从微机直到巨型机,已独立于 UNIX 和 PDP 系统。在微机上使用的有 Microsoft C、Turbo C、Quick C 等。

为了使 C 语言标准化,美国 ANSI(American National Standard Institute) 从 1983 年

开始着手制定了C语言的标准化方案,简称ANSI C。在此以后的许多C语言的新版本都参照了此标准,这样就给C语言程序的移植创造了更有利的环境。

§2 C语言的特点

程序设计语言有许多种类,每一种类都有其特殊功能和应用范围。C语言之所以成为当前世界上最流行的几种语言之一,其特点如下:

一、语言表达能力强、通用性好

C语言是面向结构程序设计的语言,通用性好,不局限于某种机器。它可以直接处理字符、数字、地址;可以完成通常由硬件实现的算术和逻辑运算;C语言能取代汇编语言来编写各种系统软件和应用软件。例如,C语言的编译程序本身就是用C语言编写;当前广泛使用的DBASE III系列关系数据库管理系统软件也是用C语言编写的;它也可用于编写数值计算、数据处理、CAD、办公室自动化、人工智能等各种应用软件。因此,可以说几乎所有的程序设计任务均可使用C语言来完成。

二、程序书写简洁、清晰

一个C语言程序由若干个函数定义的集合构成。C语言中的函数提供了编制结构化程序的手段,使得程序结构清晰并易于阅读和维护。C语言提供了丰富的运算符,使用灵活,使程序书写简洁,且执行效率高。

三、具有丰富的数据类型和结构化的控制语句

C语言既具有常规的基本数据类型,如整型、浮点型、字符型,又可以在此基础上构造各种结构类型,如数组、结构、联合等,尤其指针类型能非常方便地对任意复杂的数据结构进行运算。同时,它的各种控制语句,如if、while、for、switch等,功能很强,足以描述和编写具有良好程序风格的程序。

四、生成的代码质量高、可移植性好

C语言中,运算符多,有些运算符直接反映了现代机器指令,可生成较短的机器代码,其生成代码效率仅比用汇编语言写的代码低10%~20%。由于用C语言描述比用汇编语言描述问题编程迅速、工作量小、可靠性好,易于调试、修改和移植,所以,C语言是编写系统软件和应用软件的理想工具。而且,C语言程序可以从某一环境不加或稍加改动就可搬到另一个完全不同的环境上运行,而汇编语言则完全依赖于机器,不可移植。

五、不配备输入/输出语句

C语言中没有提供输入/输出语句,也没有现成的访问文件的指令,所有I/O功能都是通过调用库函数来完成。C提供了大量而有效的库函数,可根据需要方便地扩充,使得C本身运行时占用存储空间少且运行效率高。

C语言的优点很多,但也有一些不足之处。具体表现在运算符优先级太多,不便于记忆,

有些还与常规习惯有些不同;类型检验太弱,安全性较差,C语言允许编写者有较大的自由度,放宽了语法检查,这就要求程序设计员在编程时更为谨慎,尽管C语言有些缺点,但仍不失为一种实用的通用程序设计语言。现在国内外都在学习和使用C语言,反过来又促进了C语言本身的不断发展。

§3 C语言程序结构

本节意在通过二个完整的C程序,介绍一下C的程序结构,使读者对C程序有一个初步的印象。程序中的各个组成部分将在以后的各章中作详细的介绍。

一、简单例子

【例 1.1】 输入圆半径,求圆面积?

/* c1-1.c */	——注释
/* 程序 c1-1.c 用于计算圆面积 */	
#define PI 3.14159	——预处理命令
#include "stdio.h"	
main()	——主函数头
{ int r;	
float area;	——说明部分
printf(" input r\n");	
scanf("%d", &r);	
area = PI*r*r;	——语句部分
printf("area = %8.3 f\n", area);	
}	
C>c1-1 ↵	
input r	
10 ↵	运行结果
area = 314.159	

本程序的作用是输入圆的半径 r , 计算圆的面积 $area$ 。其中的输入输出调用了两个库函数 `scanf` 和 `printf`。库函数的说明包含在 `stdio.h` 中, 必须在源文件的头部设置 `#include "stdio.h"` 文件包含命令。`scanf` 函数作用是输入 r 的值, 并放入 r 变量中, "&" 表示取地址, 其中的 "&r" 意指变量 r 的地址。

【例 1.2】 输入任意的一行字符串, 将字符串中的小写字母转换成大写字母并输出。

```

/* c1-2.c */
#include "stdio.h"
#define MAX 80
main( )          /* 主函数 */
{ char s[MAX];
  while (gets(s) != NULL)

```

```

        putupper(s);
    }

    putupper(str)                /* 子函数 */
    char str[ ];
    { int i;
      char c;
      for (i=0;(c=str[i])!='\0';i++)
          { c=((c>='a'&&c<='z')?(c+'A'-'a'),c);
            putchar(c);
          }
    }
}

C>c1-2 ↵
good morning! ↵
GOOD MORNING!
good bye! ↵
GOOD BYE!
12345 ↵
12345
^z ↵

```

本程序定义了两个函数：主函数 main 和被调用子函数 putupper。主函数调用库函数 gets 输入一行字符串后，调用 putupper 函数将输入的字符串逐个字符检查，把包含的小写字母转换成大写字母，再调用字符输出库函数 putchar 输出字符。

二、C 程序的结构

由上述两个例子可以看出 C 程序的基本结构及书写规则如下：

1. 函数

任何 C 程序或者仅由一个主函数(main 函数)组成，或者由一个主函数和若干个子函数组成。主函数的名字一律用 main，子函数可以是系统提供的库函数，也可以是程序员定义的函数，主函数与子函数的先后次序无关。程序执行从 main 开始。

函数定义的一般形式如下：

[函数类型]函数名(形参表列)

形参说明

{ 函数体说明部分；

执行部分

}

函数名后的形参表列可有可无，有形参则要进行形参说明；花括号“{ }”括起的部分称为函数体，函数体内说明部分在前，执行部分在后，函数体内也可以出现多个花括号(即花括号可以嵌套)，但各有不同的含义。

2. C 程序书写

C 程序书写格式自由,一行内可以写几个语句;一个语句也可以分多行书写,为了程序清晰,书写时应根据语句的作用、位置进行安排,嵌套时要向右缩排。在 C 中除了符号常数外,一般变量名等都用小写字母表示。

3. 注释

以 `/*...*/` 表示注释部分。注释不是 C 程序中的语句,对编译和执行没有什么影响,仅便于人们阅读程序,注释可以出现在程序的任何地方,但 `/*` 与 `*/` 必须成对出现,不一定在一行上。

4. 预处理

预处理是 C 语言的独特优点之一,它为编程提供了方便,便于程序阅读和移植,它以 `"#"` 标志开始,主要包括:

(1) 宏定义

程序中对符号常数的定义使用 `#define` 表示。如在例 1.1 中的 `#define PI 3.14159` 定义了 PI 为符号常数取了 3.14159,在以后程序运行过程中出现的 PI 都代表了 3.14159,习惯上符号常数名用大写,其他用小写,以示区别。

(2) 文件包含

在 C 程序中若调用了 C 的某个库函数,那么必须用 `#include` 将包含该函数说明的头文件 (`*.h`) 嵌入 C 源文件,否则因有关信息在程序中未给出说明和定义,会使编译出错。上述两例子中用了标准输入、输出库函数 `printf`, `putchar`, `scanf`, `gets` 等,其函数的说明都在文件 `"stdio.h"` 中,所以必须在源文件前面设置文件包含 `#include "stdio.h"`。必须注意包含的文件名用双引号 `" "` 或尖括号 `< >` 将其括住。

习 题

1. C 语言的发展过程如何? 从它的发展过程中可以看出什么特点?
2. 试述 C 语言的特点。
3. C 语言程序的主要结构特点是什么?
4. 你学过的高级语言程序结构与 C 语言程序有何异同?

第 2 章 C语言基本概念

在第一章中,我们从总体上给出一个C程序的基本程序结构,使读者有个概貌的了解。本章将介绍C语言程序中使用的符号、数据类型以及常用的输入输出函数,以便读者一开始就能编制简单的C程序,并在计算机上进行调试。

§1 基本语法单位

任何计算机语言都根据计算机系统特定的硬件环境,规定它自己特定的一套基本符号和标识符,C语言也不例外。

一、基本符号

1. 数字 10 个(0~9);
2. 英文字母大小写各 26 个(A~Z,a~z);
3. 特殊符号,主要用来表示运算符,它通常是由 1~2 个特殊符号组成。例如:

+ - * / % < <= > >=
== != && || ! & | ~ =
++ -- ?: << >> () [] { } ,

等等。

二、标识符

标识符只起标识作用,它用来表示常量、变量、函数、类型、语句的名字。标识符命名规则:以字母或下划线(下划线也起一个字母作用)开头,字母、数字串组成,长度一般不超过 8 个字符。在不同的系统中有不同的规定:Turbo C 中规定长度不超过 32 个字符,下划线开始的标识符由系统使用。在C语言中,大、小写字母含义敏感,例如:MAX, max, Max 均表示不相同的标识符。

在C语言中,标识符分成三类:

1. 关键字

关键字(或称保留字)用来说明某一固定含义的字,不能作它用。表 2-1 中列出了C语言中所使用的关键字。

关键字

表2-1

分类	关键字	意义	分类	关键字	意义
数据类型	char	字符类型	语句类	break	跳出循环
	const	不可修改类型		case	条件判别的常量表达式定义
	double	双精度浮点类型		continue	跳到下一次循环
	enum	枚举类型		default	不满足条件时

表2-1

分类	关键字	意 义	分类	关 键 字	意 义
数 据 类 型	float	单精度浮点类型	语 句 类	do	执行循环
	int	整型		else	if...else
	long	长整型		for	循环
	short	短整型		goto	无条件转移
	signed	带符号整型		if	条件判断
	sizeof	长度计算		return	函数返回
	struct	结构类型	存 储 类	switch	条件分支
	typedef	附加类型的定义		while	循环
	union	联合体		auto	自动变量
	unsigned	无符号整型(也可字符型)		extern	外部变量
型	void	无类型	存 储 类	register	寄存器变量
	volatile	可修改类型		static	静态变量

在 Turbo C 系统中还扩充了 asm, pascal, far, huge, near 等作为关键字, 具体使用请参阅有关手册。

2. 特定字

有特定含义的若干标识符。它们主要用在 C 语言的预处理程序中。这些标识符不是关键字, 但因具有特定含义, 建议读者不要在程序中把它们作为一般标识符随意使用, 以免造成混乱。

特定字有 define, undef, include, ifdef, ifundef, endif, line 等。

3. 一般标识符

由用户定义的标识符。根据上述构成规则, 以下列出的是合法的标识符:

a_b_c maxium length string english book

x123 ywz ABC

而以下列出的是非法标识符:

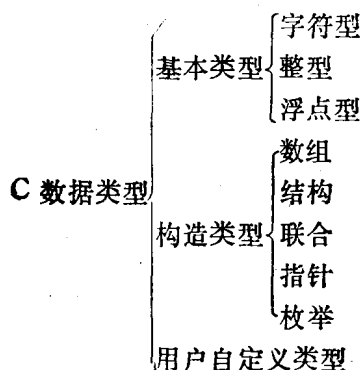
last-name 3col π a+b+c f(x)

为了使程序易读、易修改, 标识符命名应该选择恰当, 尽量符合人们习惯, 表示有意义的标识符, 一般取用英文单词、汉语拼音缩写作为标识符。需要注意的是在 C 编译系统中, 把标识符的大小写字母视为不同的符号, 一般符号常数使用大写字母, 其余均用小写字母。

§2 数据类型

在 C 语言中, 一个程序所用到的每个常量、变量和函数都隐式或显式地与一种数据类型相联系。每种数据类型表明它的可能取值和能在其上进行哪种运算。

C 语言提供了丰富的数据类型, 它们基本上可以分成三类: 基本类型、构造类型和用户自定义类型。如下所示:



这些数据类型不但可以单独用于变量的定义和说明，而且也可以按照一定的规则组合成各种更为复杂的数据类型。本章只介绍基本类型，其他类型将在以后各章中讨论。

基本数据类型(或称标准数据类型)包括字符型、整型和浮点型。表2-2中列出在IBM PC机上这三种类型的类型说明及其取值范围。对于不同的C语言系统，所支持的基本类型有多有少，而且取值范围与机器硬件有关，在使用时请查阅有关手册。

需要指出的是，C语言与一般语言不同，它没有提供布尔(逻辑)类型，在逻辑运算中，它是以非零表示真(TRUE)，以零表示假(FALSE)。

基本数据类型的存储及取值范围

表2-2

类型	符 号	占字节数	类型说明符	简 写	取 值 范 围
字 符 型	带	1	signed char	char	-128~127
	不带	1	unsigned char	—	0~255
整 型	带	2	signed int	int	-32768~32767
		2	signed short int	short	-32768~32767
		4	signed long int	long	-2147483648~2147483647
		2	unsigned int	unsigned	0~65535
	不带	2	unsigned short int	unsigned short	0~65535 (0~2 ¹⁶ -1)
		4	unsigned long int	unsigned long	0~4294967295(0~2 ³² -1)
浮 点 型	带	4	float	—	±(10 ⁻³⁸ ~10 ³⁸)7位有效位
		8	double	—	±(10 ⁻³⁰⁸ ~10 ³⁰⁸)16位有效位

注：取值范围与具体的操作系统和硬件环境有关，在IBM微机上，DOS的C语言和xenix的C语言int型分别为2字节长和4字节长

§ 3 变 量

变量是程序运行中其值可以发生变化的量。在C语言中，每个使用的变量必须预先说明，以明确它们的存储类别和数据类型。

变量说明的一般形式：

[存储类别] 数据类型 变量名表；

存储类别规定了变量在内存的存储区域和作用范围，该选择项可省，将在第七章中作详细介绍。基本数据类型见表2-2的类型说明符。

```

例如：int i, j;           /* 说明i, j是整型变量 */
      float x, y, max;     /* x, y, max是单精度数 */
      char name[20];       /* name是字符数组 */

```

变量说明是从类型说明符开头的说明语句,因此最后要以分号“;”作为结束符。

一、整型变量

由表 2-2 可知,整型分有 int、short(短整型)和 long(长整型)三种。

一般 C 编译系统的实现中,长整型变量比短整型变量所占的位数多一倍,而整型变量所占的存储空间大小和值域依赖于背景机的字长。在 16 位微机上整型占 16 位,与短整型相同,在 32 位机器上整型占 32 位,与长整型相同。

为了便于进行位运算和地址操作,整型变量还可以说明无符号(unsigned)的,无符号整型变量的值总是大于等于 0,取值范围见表 2-2。

下面是整型变量的说明:

```
int i, count;           /* 有符号整型变量 */
long sum, fac;          /* 有符号长整型变量 */
unsigned short day, month; /* 无符号短整型变量 */
```

二、浮点型变量

浮点型变量包括两种形式:float(单精度)和double(双精度)。double 占用 float 两倍的存储空间。一般单精度提供 7 位有效数字,双精度提供 16 位有效数字,当 float 型变量所具备的精度不够时,使用 double 型。

例:double area, s;
float r;

三、字符型变量

用 char 说明符说明的字符型变量占用 1 个字节的存储空间。用于存储一个表示该字符的整数值。一般字符的整数取自 ASCII 中对应的字符代码值,也有的机器用的是 EBCDIC 字符集。不管哪种字符集,都具有下述性质:

1. 每个字符必须有一个不同的字符代码值。
2. 数字字符 '0' < '9'、小写字母 'a' < 'z'、大写字母 'A' < 'Z',即按序排列,大、小写字母的编码不一样。
3. 字符型变量也可作为整数型变量取其码值在表达式中参加运算。

下面是字符型变量的说明:

```
char c1, c2, a1, a2;
```

表示 c1, c2, a1, a2 为字符型变量;各可存放一个字符。

在 C 语言中如何存储字符串呢?这关系到字符数组。有关内容将在第五章中介绍,为以后举例方便,这里先对字符数组作一简单介绍,对字符数组的说明是:

```
char 字符数组名[整型表达式];
```

整型表达式可以是常数、符号常数,用以说明字符数组的大小。

例:char name[20];

表示 name 是个字符数组,含有 20 个元素,每个元素放一个字符。C 语言规定,对有 n 个元素的数组,数组元素下标从 0 开始,直到 n-1。所以 name 数组的各元素为

name[0], name[1], ..., name[19]

必须注意的是:字符数组的最后一个元素必须放字符串结束符'\0',以确定字符数组的实际长度,并为处理字符串带来方便。

§4 常 量

在程序运行中,值不允许发生变化的量称为常量。与变量相对应,常量有整型常量、浮点型常量、字符型常量和字符串常量。

一、整型常量

整型常量由一个或多个数字序列组成,中间不允许用逗号分隔。可以使用十进制常量、八进制常量和十六进制常量。

1. 十进制常量

形式: $\pm n$

其中 n 是 0~9 的数字序列,最高位不能是 0, "+" 为正,可省, "-" 为负。

例: 1234, -1000, -1

表示十进制常量。

而 1,234 10² 10/3 0123

是非法的十进制常量。

2. 八进制常量

形式: $\pm 0n$ /*0 是数字 0,而不是字母 O*/

其中 0 表示八进制数的引导符,不能省, n 是 0~7 的数字序列。

例: 0123, 01000, -01

表示八进制常量。

而 01289, 123, 05,670

是非法的八进制常量。

3. 十六进制常量

形式: $\pm 0xn$

其中 0x 表示十六进制数的引导符,不能省。 n 是 0~9, a~f 或 A~F 的数字、字母序列。一般前面的 x 字母小写,后面的 'a'~'f' 也应小写, 否则反之。 'a'~'f' 或 'A'~'F' 表示数值 10~15。

例: 0x12c, 0x100, 0XFFFF

表示十六进制常量。

对于超过取值范围的整数,可以在数字后加字母 L 或 l,表示长整型常量。

例: 123456L, 0XFFFFFFL

二、浮点型常量

C 语言中的浮点型常量如同一般语言中的实数,所有的浮点型常量都看作为双精度类型。浮点型常量有两种表示形式:小数形式和指数形式。C 语言的浮点型常量与 FORTRAN