



C

语言教程

郑丽英 编著

兰州大学出版社

C 语言 教 程

郑丽英 编著

内容提要

C语言是近年来发展、推广最快的一种通用高级程序设计语言,它具有功能强、效率高、简洁灵活、可移植性好等特点,在软件工程领域里越来越受到人们的普遍重视,其应用日益广泛。

本书较全面、系统地介绍了C语言及其程序设计方法和应用。全书共分八章,主要内容包括C语言概述、基本运算符和表达式、控制语句、数组、函数、结构、指针、输入输出和文件处理。

本书深入浅出,通俗易懂,内容充实,实用性强。读书是一本较好的学习和掌握C语言及其应用的教科书,可做为高等院校计算机专业和非计算机专业的教科书,也可供参加计算机等级考试以及各部门的计算机工作者、科研、工程技术人员学习参考。

C语言教程

郑丽英 编著

兰州大学出版社出版发行

兰州市天水路 216 号 邮编:730000 电话:8617156

静宁县印刷厂印刷

开本:787×1092 毫米 1/16 印张:18.75

1996 年 1 月第 1 版 1996 年 1 月第 1 次印刷

字数:456 千字 印数:1—8000 册

ISBN7-311-00967-7/T·27 定价:25.00 元

前 言

C 语言是贝尔实验室于 70 年代初期在已开发的程序设计语言基础上发展起来的,它与 UNIX 操作系统的发展有着密切的关系。UNIX 操作系统早期是用汇编语言编写的,后来用 C 语言重写,使新的操作系统的规模仅为旧的三分之一。这一尝试的成功,确信 C 语言是一种有效的程序设计工具。UNIX 操作系统的普及促进了 C 语言的发展和推广。由于 C 语言有着经济实用的表达式、丰富的控制流和数据结构以及语言简单、灵活、易于实现等特点,因此 C 语言在大型机、中型机、小型机以及微型机上都得到实现,不仅 UNIX 操作系统支撑的计算机上有 C 语言,而且在许多非 UNIX 操作系统支撑的计算机上也都实现了 C 语言,使 C 语言成为一种当今主流程序设计语言。

C 语言既是系统语言,也是很好的应用语言。它已有一个很大的函数库,可在各种应用类型中发挥作用;也可以为某种特殊要求的课题编制函数并放入函数库中,以扩充系统的功能。

C 语言是通用的程序设计语言,它不针对任何具体的机器,语言本身考虑到了可移植性,因此有着良好的可移植的性能。

C 语言容易学习,容易编写。用 C 语言编制的程序清晰紧凑,有助于模块化和好的程序结构。

C 语言在提高目标程序的效率方面作了努力,使编译出的目标质量好,适应于编制各种系统软件、应用软件,是程序设计的一种理想的语言工具。

本书共分八章。

第一章简单论述 C 语言的发展历史、特点、程序结构等,同时还结合简单例子对 C 语言的编写、编译和运行做了介绍。

第二章介绍 C 语言的基本概念、成份和表达式。表达式是语言的基本组成部分,本章对各种主要表达式作了介绍。此外还介绍了 C 语言的运算符,列表给出它们的优先顺序。

第三章介绍 C 语言的各种控制语句,并给出大量的例子阐述了各种控制语句的使用方法。

第四章介绍 C 语言的数组。在这一章还给出四种常用的排序算法的实现。

第五章介绍 C 语言的主要成份——函数,论述 C 语言的函数结构、函数类型、函数参数、函数的嵌套调用以及函数的递归调用,还介绍了 C 语言中很有特色的变量存储类。

第六章介绍 C 语言中最有特色的内容——指针,详细论述指针的基本概念、指针与数组的关系,指针数组、函数指针以及命令行参数等内容,这一章的内容是学好 C 语言并付诸实践的关键部分。

第七章介绍结构和联合,包括结构的定义、结构成员的引用、结构数组、指向结构的指针,还介绍了引用自身的结构以及链表的概念和对链表进行操作的各种算法。

第八章介绍输入输出以及文件的使用,这些内容虽然不是 C 语言成份,但却是编写 C 语言程序所必需的知识,一个有用的 C 语言程序是少不了这些内容的。

本书在编写过程中,兰州大学计算机系席先觉教授以及兰州铁道学院电信系时天保教授为提供了非常有益的建议,席先觉教授还仔细审阅了手稿,提出了宝贵的意见,借此表示衷心的感谢。

由于时间仓促,加之本人水平有限,书中缺点有所难免,恳望得到批评和指教。

编者

1995 年 6 月兰州

序

随着我国国民经济信息化时代的到来,计算机的应用已在我国各个领域得到普及。尤其是三金工程(即我国的信息高速公路基础设施)的开发应用,国内多个全国性网络与美国互联网(Internet)的开通,实现了与国际信息高速公路的“接轨”,更显示出计算机应用的迫切性和重要性。目前,以计算机技术为核心的新技术的广泛应用,正在开始改变着人们的生产方式、工作方式、学习方式和生活方式。因此,计算机的基本知识和应用能力,已成为从事各行业人员必须具备的基本文化知识(有人称之为“第二文化”),致使很多行业在职员职称晋升和聘用从业人员时,把计算机应用能力作为考核的基本内容,国家人事部于今年明确规定,从96年起,工程技术、自然科学研究、高校理工科教师、农业、统计等几个专业晋升职称时必须通过计算机考核。

为了适应形势的需要,国内各高等院校各类专业对计算机的教学都非常重视,很多省市的各级教育部门,为了促进各自高校非计算机专业的计算机教学工作,提高计算机课程的教学质量,考核在校大学生计算机方面的素质,都分别组织了“计算机基础知识和应用能力”的考试。

国家教委考试中心遵照中央关于为社会主义经济建设服务,打破部门所有,条块分割、拓宽社会服务领域的精神,于去年推出全国计算机等级考试,为全国各行业、各单位录用与考核工作人员,提供一个科学、统一、公正的基础性考试。去年开考以后,在国内外都引起了热烈的反响。

基于同样的目的,甘肃省教委和人事局也于去年10月组织了全省高校大学生的“计算机基础知识和应用能力等级考试”,并决定于今年推向社会,欢迎社会考生应考。为了鼓励在校大学生和社会从业人员学习计算机技术,对参加该考试并取得合格证书者,晋升职称时可以免考或在同等条件下优先考虑。求业人员可在同等条件下优先录用,并严格规定自97年起大学毕业生都须取得该考试的合格证书才能分配到省级党政机关、高等学校和科研单位工作。

为了帮助和指导广大考生学好计算机基础知识,较好地掌握计算机技术,顺利通过“计算机基础知识和应用能力等级考试”,适应学习与工作需要,在省教委和甘肃省高校计算机基础教育研究会及兰州大学出版社的支持下,成立了“计算机等级考试系列丛书编委会”,组织编写计算机等级考试指导教材,供广大考生参考使用。

我们深信,在省教委的热情关怀和支持下,经过编委会同志和我省各高等学校计算机专家和老师们共同的辛勤努力,一定能使我省高校计算机教学水平有一个很大的提高,呈现出一个生动活泼、欣欣向荣的可喜形势。

编委会

一九九五年十月

目 录

第一章	C 语言概述.....	(1)
§ 1.1	C 语言的发展演变历史	(1)
§ 1.2	C 语言的特点	(1)
§ 1.3	C 语言程序的基本结构	(2)
§ 1.4	C 语言程序的编译和运行	(5)
第二章	数据类型、运算符和表达式	(8)
§ 2.1	标识符和变量定义	(8)
§ 2.2	常量	(9)
§ 2.3	基本数据类型及其转换	(13)
§ 2.4	运算符和表达式	(19)
§ 2.5	例题解析	(33)
习题		(34)
第三章	控制语句	(36)
§ 3.1	条件语句	(37)
§ 3.2	switch 语句	(44)
§ 3.3	循环语句	(48)
§ 3.4	中断、继续、转向、返回语句	(59)
§ 3.5	例题解析	(63)
习题		(70)
第四章	数组	(72)
§ 4.1	一维数组的定义和使用	(72)
§ 4.2	二维数组的定义和使用	(74)
§ 4.3	字符数组	(78)
§ 4.4	基本排序算法的实现	(84)
§ 4.5	例题解析	(88)
习题		(96)
第五章	函数和程序结构	(97)
§ 5.1	函数	(97)
§ 5.2	变量的存贮类别及作用域规则	(113)
§ 5.3	变量的初始化和分程序结构	(126)
§ 5.4	C 预处理程序	(127)
习题		(130)
第六章	指针	(132)
§ 6.1	指针的基本概念	(132)

§ 6.2	指针和数组	(135)
§ 6.3	指针数组和多级指针	(141)
§ 6.4	命令行参数	(144)
§ 6.5	指向函数的指针	(148)
§ 6.6	例题解析	(150)
习题		(157)
第七章	结构和联合	(159)
§ 7.1	结构的基本概念	(159)
§ 7.2	结构和函数	(166)
§ 7.3	结构和数组	(170)
§ 7.4	引用自身的结构	(182)
§ 7.5	联合	(195)
§ 7.6	枚举类型	(199)
§ 7.7	定义类型	(201)
§ 7.8	字段	(203)
§ 7.9	例题解析	(206)
习题		(213)
第八章	输入输出和文件处理	(215)
§ 8.1	输入输出函数	(215)
§ 8.2	文件操作	(224)
§ 8.3	输入输出重定向和错误处理	(236)
§ 8.4	文件的随机访问	(239)
§ 8.5	非缓冲文件系统	(243)
§ 8.6	例题解析	(246)
习题		(251)
附录 I	1994 年全国计算机等级考试二级 C 语言笔试试题	(253)
	甘肃省计算机等级考试 C 语言笔试试题	(266)
附录 II	常用字符与 ASCII 代码对照表	(272)
附录 III	TURBO C 实用库函数	(275)
附录 IV	C 语言中的关键字	(287)
附录 V	计算机等级考试 C 语言试题参考答案	(288)

第一章 C 语言概述

§ 1.1 C 语言的发展演变历史

C 语言是在 B 语言的基础上发展起来的,它的根源可以追溯到 ALGOL 60. 1960 年出现的 ALGOL 60 是一种面向问题的高级语言,它过于抽象,离硬件比较远,不宜用来编写系统程序。1963 年,英国的剑桥大学推出了 CPL 语言(Combined Programming),CPL 语言是在 ALGOL 60 的基础上发展起来的,相比较更接近硬件一些,但仍然过于庞大和复杂,难以学习和编程。1967 年英国剑桥大学的 Martin Richards 对 CPL 语言作了简化,推出了 BCPL (Basic Combined Programming Language) 语言,它吸取了 CPL 的一些精华。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,又作了进一步简化,设计出了很简单且很接近硬件的 B 语言(取 BCPL 的第一个字母),并用 B 语言写了第一个 UNIX 操作系统,在 PDP-7 上实现。1971 年,在 PDP-11/20 上实现了 B 语言,并写了 UNIX 操作系统。但 B 语言过于简单,功能有限,并且由于 PDP-11 机器是以字节编址的,而 B 语言面向字,这样妨碍了对单个字节进行存取的能力。1972 年至 1973 年间,贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母)。C 语言既保持了 BCPL 和 B 语言的优点(精炼、接近硬件),又克服了它们的缺点(过于简单,数据类型单一等)。此后,C 语言多次作了改进,并用 C 语言重新编写了 UNIX 操作系统。用 C 语言书写的 UNIX 操作系统比原先的版本更易于理解、修改、扩充。特别地,具有很好的移植性。

虽然,最初的 C 语言是附属属于 UNIX 系统且在 PDP-11 机上实现的,但随着 UNIX 系统的广泛应用,C 语言开始为人们普遍接受。现在,C 语言已独立于 UNIX 系统,独立于 PDP-11 机器而蓬勃发展。1978 年以后,C 语言已先后移植到大、中、小型微型机上,成为世界上应用最广泛的几种计算机语言之一。

§ 1.2 C 语言的特点

C 语言具有许多特点,归纳起来主要有以下几个:

1. 语言简洁 C 语言是一种小型语言,它一共只有 32 个关键字,9 种控制语句。C 语言采用的表示法尽量简洁,如以 { } 代替通常的 begin 和 end 作复合语句或函数体的“括号”。以 ++ 表示加 1,-- 表示减 1 等等。C 语言本身不提供输入输出设施,也没有并行操作、同步或协同程序等复杂的控制结构,这些成份由支持 C 语言的系统环境提供,通过显式函数调用来实现。

2. 表达能力强 C 语言可以直接处理字符、数字、地址,能进行位 (bit) 操作,可以完

成通常要由机器指令来实现的普通的算术及逻辑运算。因此它面向机器可以代替汇编语言来编写各种系统软件和应用软件。最明显的例子就是：UNIX 是用 C 语言编的，此外还有一些数据库管理系统如 INGRES、INFORMIX、DBASE II 以及 Logo, Prolog 解释系统等都是由 C 语言描写的，这表明 C 语言是开发软件的有力工具。

3. 运算符丰富 C 语言的运算符包含的范围很广泛，共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其它高级语言中难以实现的运算。

4. 数据类型丰富 C 语言的数据类型有：整型、实型、字符型、数组类型、指针类型、结构类型、联合类型等。能用来实现各种复杂的数据结构（如链表、树、栈等）的运算。尤其是指针类型数据，使用起来比 PASCAL 更为灵活、多样。另外它的各种控制语句（如 if... else 语句、while 语句、do-while 语句、switch 语句、for 语句）功能很强，足以描述结构良好的程序。

5. 生成代码质量高，程序执行效率高。一般只比汇编程序生成的目标代码效率低 10~20%。

6. 可移植性好 用 C 语言写的程序基本上不作修改就能用于各种型号的计算机和各种操作系统。

§ 1.3 C 语言程序的基本结构

C 语言程序一般由若干个函数组成，而这些函数可以保存在一个或几个源程序文件中，这些文件都是以 C 为文件扩展名。

一、简单的示例

例 1.1

```
main( )
{
    Printf("This is a C Program. \n");
}
```

其中 main 表示“主函数”，每一个 C 程序都有一个 main 函数。这个函数中只有一条语句，Printf 是一个打印函数，它在屏幕上显示括号内的字符串。反斜线\可与后随的一个字符合并在一起代表一个字符，如\n 表示换行，\t 表示水平制表符（即符号 HT）。

例 1.2

```
main( )
{
    int a,b,sum;      /* 定义变量 a,b,sum */
    a=1;
    b=2;
    sum=a+b;          /* 求 a 与 b 的和 */
    printf("a ADD b is %d",sum);
}
```

这个程序只有一个 main 函数。函数中第一个语句定义变量 a、b 和 sum 为整型变量。/*....*/表示注释部分,注释部分可以出现在 C 程序的任何部位,它只是给人看的,对编译和运行不起作用。第二、三、四个语句是给变量赋初值并进行求和,主函数体内的最后一条语句,是对 Printf 函数的调用。例 1.2 比例 1.1 的情况稍微复杂一些,这里出现了两个参数,第一个参数是字符串,第二个是变量 sum。第一个字符串提供格式信息,在“%”后面出现的字符是格式转换符,指出输出其它(第二个、第三个、...)参数的格式,其它参数就是按指定格式输出的内容。字符串内其它字符,按原样打印。本例中的格式转换 %d 表示下一个参数(即第二个参数 sum)将作为一个十进制数打印出来。因此程序的执行结果是: a ADD b is 3.

例 1.3

```
main( )           /* 主函数 */
{
    int a,b,c;      /* 定义变量 */
    scanf(" %d,%d",&a,&b); /* 输入变量 a 和 b 的值 */
    c=SUM(a,b);     /* 调用 SUM 函数,将得到的值赋给 c */
    printf("SUM is %d",c); /* 输出 c 的值 */
}

int SUM(x,y)       /* 定义函数 SUM,函数值为整型,x 和 y 为形参 */
int x,y;           /* 对形参 x 和 y 的类型进行说明 */
{
    int s;          /* 定义函数体中要用到的变量 */
    s=x+y;
    return(s);      /* 将 s 的值返回 */
}
```

本程序有两个函数: main 函数(主函数)和 SUM 函数(非主函数)。主函数 main 从终端得到两个数据 a 和 b,调用函数 SUM,并将 a 和 b 传送给 SUM,在函数 SUM 中对 a 和 b 求和,将值返回给主函数,然后在主函数中进行输出。这只是一个简单的函数调用的例子,对于有关的详细概念会在后面的有关章节给出介绍。

另外,程序中的 scanf 函数其调用方法类似于 printf 的调用,它按第一个参数指定的格式输入数据给后面的参数,其格式转换的意义和 printf 函数中的格式转换很相似,%d 表示按十进制输入数据。注意,对于数值型数据不可省略记号“&”,如例中的 &a 和 &b。

应该指出的是,关于数字的输出还可以指定该数所占的字符位置以及小数部分的位数。例如,%10d 表示打印的十进制数至少占 10 个字符位置;%10f 表示打印的十进制浮点数至少占 10 个字符位置;%.5f 表示要打印 5 位小数,%10.5f 表示打印的十进制浮点数至少占 10 个字符位置;小数部分占 5 位,扣除小数点的位置,整数部分至少占 4 个字符位置。表 1-1 列出了常用的一些格式转换。

表 1-1 常用格式转换

格式串	printf 中相应的参数 输出形式	scanf 中相应的参数 输入形式
%d(%Ld)	十进制(长)整数	十进制(长)整数
%u(%Lu)	十进制无符号(长)整数	十进制无符号(长)整数
%o(%Lo)	八进制(长)整数	八进制(长)整数
%x(%Lx)	十六进制(长)整数	十六进制(长)整数
%f	十进制浮点数	十进制浮点数
%lf		十进制双精度浮点数
%g	同%f,除去不必要的0	
%c	单个字符	单个字符
%s	字符串	字符串

二、C 语言程序的结构特性

1. C 语言程序是由函数构成的。一个 C 语言源程序至少包含一个函数(即 main 函数),也可以包含一个 main 函数和若干个其它函数。这里的函数相当于其它语言中的子程序概念,用于实现特定的功能。函数可以分为两大类:一类是系统提供的库函数(例如 printf 和 scanf 函数)。另一类是用户根据需要自己编制设计的函数(如例 1.3 中的 SUM 函数)。一个 C 语言程序无论有多少个函数,无论函数出现的顺序如何,程序中有且仅有一个函数,它的名字为 main 即主函数,并且程序总是从 main 函数开始执行。

2. 一个函数由两部分组成:

(1)函数头 函数头是函数的定义部分。一般格式为:

函数类型说明符 函数名(参数表)

参数说明语句;

其中参数表可为空,参数表为空时,函数名后面的一对圆括号不能少。如例 1.3 中 SUM 函数的函数头为:

```
int SUM(x,y)
```

```
int x,y;
```

(2)函数体 函数体由括号{}括起来的一系列语句构成。函数体中通常包括变量定义语句和可执行语句。有时候函数体中既无变量定义也无可执行部分,此时函数是一个空函数,例如 dump() {...} 是一个空函数,什么也不干,但确实是合法的 C 语言函数。空函数主要用于编制大型软件时扩充函数功能。在程序设计中往往根据需要确定若干模块,分别由一些函数来实现。在第一阶段只设计最基本的模块,其它一些次要功能或锦上添花的功能则在以后需要时陆续补上。在编写程序的开始阶段,可以在将来准备扩充功能的地方写上一个空函数,函数名取将来采用的实际函数名,只是这些函数未编好,先占一个位置,以后用一个编好的函数代替它。这样做,程序的结构清楚,可读性好,以后扩充新功能方便,对程序结构影响不大。空函数在程序设计中经常使用。

3. C 语言程序书写格式自由 C 语言程序中,一行内可以写几个语句,一个语句也可以写在多行上。C 语言程序中语句没有行号。

4. 分号是 C 程序的语句终止符 每一个数据定义和可执行语句的最后必须有一个

分号,即使是程序的最后一个语句也应该有一个分号(PASCAL 中分号是语句间隔符,最后一个语句可省略分号)。

§ 1.4 C 语言程序的编译和运行

程序员编好了 C 语言源程序后,需要使用某一个版本的 C 语言编译系统对源程序编译以产生一个可以执行的目标文件,再运行这个程序(即目标文件)就可获得所期望的结果。

在不同的运行环境下,使用不同的编译系统运行 C 语言程序的过程(使用的命令)是不相同的。下面我们简单介绍三种不同环境下运行 C 语言程序的步骤。

一、在 DOS 下用 MSC4.0 编译程序运行 C 程序的步骤。

1. 编辑 C 语言源程序。可以用行编辑程序 EDLIN 或 WORDSTAR 对 C 语言源程序进行编辑。这一步将产生 C 语言源文件。在一般情况下,一个 C 语言程序可以放在一个文件中,有时可放在若干个文件中,这些文件均要以 .C 作为文件扩展名,比如 example1.c, example2.c…。文件以 .c 作为扩展名表示这是 C 语言源文件。

2. 编译。要经过 4 道编译。假设某一 C 语言源文件名为 example.c, 则编译过程为:

```
cc1    example.c    (第一次编译,作用是预处理)
cc2    example.c    (第二次编译,作用是语法分析)
cc3    example.c    (第三次,编译)
cc4    example.c    (第四次,优化编译)
```

如果在编译过程中出现出错信息,需重新编辑(修改)源程序,然后再编译,这一过程也许要重复多次,直到编译通过,无任何错误为止。

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 example,,,cslib
```

cslib 是 C 系统库 (C System Librarg) 的缩写。链接成功将产生名为 example.exe 的可执行文件。

4. 执行程序。只需输入目标文件名(不带后缀)。

```
example
```

二、在 UNIX 操作系统下运行 C 程序的步骤

1. 编辑 C 语言源程序。可以用 UNIX 系统的文本行编辑程序 ed, 或屏幕编辑程序 vi 对源程序进行编辑。

2. 编译。调用 C 语言编译程序 cc 对源文件进行编译。可打入命令:

```
cc example.c
```

可以看到一般的编译命令是

```
cc 文件名
```

打入这个命令后,系统把 C 语言编译程序调入内存对要编译的程序进行词法分析、语法分析、代码生成等多种处理。如果编译有错,则需要重新编辑、编译直到无错。如果没有错,就按缺省原则,把最后的可执行目标程序放在文件 a.out 中。这里 a.out 是系统指定的目标文件名。

3. 执行程序。只要打入命令

a.out

如果不想用系统指定的文件名 a.out 作为可执行文件名,也可以在编译时自己指定可执行文件名。例如想指定为 example.out,可以在编译时打入如下命令:

```
cc -o example.out example.c
```

它的作用是将源程序 example.c 编译成可执行目标程序,把它放在 example.out 文件中。此时,a.out 中不存放目标文件。执行时可打入命令.

```
example.out
```

三、用 Turbo C 运行 C 语言程序的步骤

1. 首先将 Turbo C 装入磁盘

2. 调用 Turbo C 程序,只需打入:

```
tc
```

屏幕顶部出现一排“命令”行菜单(主菜单): FILE EDIT RUN COMPLIER PROJECT OPTIONS DEBUG 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。这一行上的命令每一个都能下拉出一个子菜单。例如,对于命令 FLLE,表示对文件进行操作,包括建立、保存、调入、改变目录等等。相应的下拉式子菜单为:

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

可用箭头“↓”和“↑”进行选择合适的命令,也可以键入热键(例如,与 Load 相应的热键为 F3)进行选择。其中“New”表示建立新文件,可以用于输入源程序。“Load”用于调入已存在的文件。“Save”用于保存文件到磁盘上。

3. 编辑源文件。根据需要输入或修改源程序。

4. 编译源程序。只要按“F9”键即可进行编译,链接,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,屏幕上显示源程序,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。根据这些信息可以修改源程序。再按“F9”进行编译,如此重复直到编译无错。

5. 执行程序。按“F10”键,回到主菜单。用“→”和“←”键将光标移到“RUN”处,按回车键,即可执行编译好的目标文件。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,并用箭头“←”或“→”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。

6. 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。

第二章 数据类型、运算符与表达式

§ 2.1 标识符和变量定义

2.1.1 标识符

标识符是一个字符序列,标识符用来标记常量、变量、数据类型、函数及程序的名字。

在 C 语言中构成标识符必须符合下列语法规则:

1)以字母或下划线符中任一字符打头;

2)在第一个字符之后,可以是任意的字母、下划线或数字组成的字符序列,这个序列可以是空串。

下面是合法的标识符,也是合法的变量名。

`max, sum, new, NEW, _abc, _102, L_1_2_3, A, B, x10。`

下面是不合法的标识符:

`3A, A1-4*, #500`

要注意的是,C 语言是区分大小写字母的。即大写字母和小写字母被认为是两个不同的字符。因此,NEW 和 new 是两个不同的标识符。关于标识符的长度(字符个数),C 语言无统一规定,随系统而不同,但至少应是一个字母,最多不要超过 8 个字符,假如程序中出现的变量名长度大于 8 个字符,则只有前面 8 个字符有效,后面的不被识别。例如,有两个变量:student_number 和 student_name,由于二者的前 8 个字符相同,系统认为这两个变量是一回事而不加区别。可以将它们改为:stu_name 和 stu_num,以使之区别。这种错误在编译时是不会报错的,但运行结果显然不对。

C 语言中的标识符可分为三类

1. 关键字 关键字是 C 编译程序预先登录的标识符,共有 32 个,是语言特有的成份,它们代表固定的意义,不能做一般标识符使用。这些关键字都是用小写字母构成的。附录 IV 给出了 C 语言中的所有关键字。

2. 特定字 这些是具有特定含义的标识符。这些标识符虽不是关键字,但是人们已习惯把它们看成是关键字,并赋予了特定含义。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。这些特定字是:

`define, include, undef, ifdef, indef, endif` 和 `line` 它们主要用在预处理程序中。

3. 一般标识符(用户定义字) 是用户根据标识符的构成规则定义的标识符。这样的标识符数目不限,需要多少定义多少,本身没有固定的含义,只有经过定义说明之后,才具有

特定的含义。不过,习惯上在定义标识符时,最好与被命名的对象有一定的联系,这样比较醒目,例如,SUM 表示求和的意思,max 表示取最大值。这样可以增加程序的易读性。

2.1.2 变量定义

其值可以改变的量称为变量,一个变量应该有一个名字(标识符),在内存中占据一定的存储单元,在该存储单元中存放变量的值。特别地,要求对程序中所有要用到的变量在使用之前必须显式地被定义。即“先定义,后使用”。定义变量就是说明变量属于什么数据类型。这样使编译程序能为变量分配相应的存储单元,且检查该变量所进行的运算是否合法。如果定义 x 是整型变量,则编译程序将为 x 分配一个 16 位的存储单元;若 x 为字符型变量,则将得到一个 8 位的存储单元。显然,不同数据类型的变量将占据不同长度的存储单元。因此,变量的定义是十分重要的。在 C 语言中变量的定义格式如下:

类型区分符 变量名表列

例如

```
int    a,b,sum;
char   ch,d;
float  e,f;
```

分别定义 a,b,sum 是整型变量, ch 和 d 是字符型变量, e 和 f 是浮点型(实数)变量。

§ 2.2 常 量

在程序运行过程中,其值不能被改变的量称为常量。C 语言中有多种常量,下面我们分别介绍。

2.2.1 整型常量

C 语言中整型常量分十进制、八进制、十六进制。

1. 十进制整数

十进制整数是带有正负号(±),由数字 0~9 组成的数字序列。如 12, -17, 0 都是有效的整型常量。数字之间不允许有空格或其它标点符号(如逗号)。并且十进制整型常量不能以 0 开头,如 0129 是不合法的。这是为了区别于八进制数。

2. 八进制整数

八进制整数是以数字 0 开头,后随数字 0~7 组成的数字序列。如 0123, 011, 07, 050 都是合法的八进制整数,分别表示十进制的 83, 9, 7, 40。而 0189, 178 则不是合法的八进制整数。由于格式符“%0”表示按无符号八进制形式输出,所以,通常认为八进制数是无符号数。有的 C 语言书中明确规定八进制数为无符号数。

3. 十六进制整数

十六进制整数是以数字 0 和字母 x 开头,由数字 0~9、字母 a~f (大小写均可)组成的数字字母序列。其中字母 a~f 分别代表 10~15 的数值。如

0x12, 0x2f, 0xa5。

均为十六进制整数常量,它们分别代表十进制数 18, 47 和 21。