



21世纪高职高专规划教材·计算机系列

C语言程序设计

陈孟建 编著

2C-43



電子工業出版社

<http://www.phei.com.cn>

767

21 世纪高职高专规划教材·计算机系列

C 语言程序设计

陈孟建 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本通用的 C 语言程序设计实用教材,编写格式由理论知识、课后练习、实训内容等三部分组成。采取循序渐进的内容安排,通俗易懂的讲解方法,书中采用的应用实例大多是当前计算机语言中使用较为广泛且读者比较喜欢的,具有通俗性、新颖性和实用性,特别适合高职高专院校培养学生的动手能力。

本书包括 C 语言程序设计基础,数据描述与基本操作,结构化程序设计,模块化程序设计,指针、结构体、共用体和枚举、文件、C 语言综合训练等内容。

本书可作为高职高专院校 C 语言程序设计教材或参考用书,也适合具有中等以上文化程度的读者自学之用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C 语言程序设计/陈孟建编著. —北京:电子工业出版社,2002.5

21 世纪高职高专规划教材. 计算机系列

ISBN 7-5053-7672-1

I . C… II . 陈… 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(2002)第 034108 号

责任编辑:束传政 特约编辑:逢积仁

印 刷 者:北京市增富印刷有限责任公司

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:20.25 字数:538 千字

版 次:2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

印 数:6 000 册 定价:24.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。

联系电话:(010)68279077

前 言

本书是一本通用的 C 语言程序设计实用教材,编写格式是由理论知识、课后练习、实验内容等三部分组成。采取循序渐进的内容安排,通俗易懂的讲解方法,书中采用的应用实例大多是当前计算机语言中使用较为广泛且读者比较喜欢的,具有通俗性、新颖性和实用性,特别适合高职高专院校培养学生的动手能力。

本书的特点是:

1. 本书自始至终贯穿两条主线,即 C 语言的学科主线和实例主线,其中 C 语言的学科主线是我们编写的主线,实例主线是学生学习 C 语言的主线。

2. 本书在写法上有别于传统教科书,突出高等职业技术学院的操作性、技能性,强调实例教学,实训教学。

3. 本书在编写结构中,采用先提出问题,再对该问题进行分析,然后给出解决问题的方法,最后通过对实例举一反三,使学生增强分析问题和解决问题的能力。每章均有强调技能提高的内容,例如进阶知识部分。

4. 本书所有的例子均可在 Turbo 2.0 以上运行通过,有较为详细的上机操作过程。

本书可作为高职高专院校 C 语言程序设计教材或参考用书,也适合具有中等以上文化程度的读者自学之用。

本书在编写过程中,得到了周铁根、陈惠玲、沈美莉、刘逸平、张寅利、田文雅、商玮等专家、教授们的帮助,在此表示衷心的感谢!

由于写作时间仓促和作者水平有限,书中不当之处在所难免,敬请读者批评指正。

编 者

2002 年 3 月于杭州

目 录

第 1 章 C 语言程序设计基础	(1)
1.1 一个大家熟悉的例子	(1)
1.1.1 一个简单的 C 语言程序例子	(1)
1.1.2 程序的简单分析	(2)
1.1.3 C 语言程序结构	(3)
1.1.4 字符及字符集	(6)
1.2 C 语言的由来、特点和发展	(8)
1.3 输入与输出函数	(9)
1.3.1 scanf 函数(标准格式输入函数)	(9)
1.3.2 printf 函数(标准格式输出函数)	(11)
1.3.3 getchar 函数(单个字符输入函数)	(15)
1.3.4 putchar 函数(单个字符输出函数)	(16)
1.4 C 语言程序的实现	(18)
1.4.1 C 语言程序实现方框图	(18)
1.4.2 C 语言的编辑	(19)
1.4.3 编译与运行程序	(20)
1.4.4 C 语言程序错误分类	(21)
1.5 进阶部分知识	(23)
1.5.1 转义字符	(23)
1.5.2 scanf 函数格式说明符	(23)
1.5.3 printf 函数格式说明符	(24)
1.6 习题一	(29)
1.7 实训一	(30)
第 2 章 数据描述与基本操作	(33)
2.1 数据类型	(33)
2.1.1 一个 C 语言程序例子	(33)
2.1.2 数据类型	(36)
2.1.3 数据的存储格式、范围与有效位	(36)
2.1.4 赋值语句	(39)
2.2 常量与变量	(40)
2.2.1 常量	(40)
2.2.2 变量	(43)
2.2.3 变量说明语句	(44)
2.2.4 各种数值型数据间的混合运算	(44)
2.3 运算符与表达式	(45)

2.3.1 C 语言的运算符	(45)
2.3.2 算术运算符	(46)
2.3.3 算术运算符的优先级	(48)
2.3.4 算术表达式	(49)
2.4 关系运算、逻辑运算与条件运算	(51)
2.4.1 关系运算及表达式	(51)
2.4.2 逻辑运算及表达式	(52)
2.4.3 条件运算	(55)
2.5 进阶部分知识	(55)
2.5.1 赋值表达式	(55)
2.5.2 赋值中的类型转换	(57)
2.5.3 变量初始化	(58)
2.5.4 逗号运算符	(59)
2.6 习题二	(60)
2.7 实训二	(62)
第 3 章 结构化程序设计	(65)
3.1 结构化程序设计思想	(65)
3.1.1 一个 C 语言程序例子	(65)
3.1.2 结构化程序设计由来	(68)
3.1.3 结构化程序分析方法	(69)
3.1.4 结构化系统分析方法的优点	(70)
3.2 程序设计工具	(70)
3.2.1 算法	(70)
3.2.2 程序流程	(71)
3.2.3 盒图(N S)	(72)
3.2.4 PAD 图	(73)
3.2.5 结构图(SC)	(74)
3.3 选择结构程序设计	(75)
3.3.1 简单条件分支语句	(75)
3.3.2 选择条件分支语句	(78)
3.3.3 无条件转向语句	(80)
3.4 循环结构程序设计	(82)
3.4.1 循环结构程序概念	(82)
3.4.2 while 结构	(85)
3.4.3 do-while 结构	(86)
3.4.4 for 结构	(87)
3.4.5 双重循环结构	(90)
3.4.6 break, continue 及 exit 语句	(92)
3.4.7 几种循环的比较	(96)
3.5 进阶部分知识	(96)

3.5.1	条件分支语句的嵌套	(96)
3.5.2	开关语句	(99)
3.5.3	循环语句的嵌套	(102)
3.6	习题三	(105)
3.7	实训三	(107)
第 4 章	模块化程序设计	(112)
4.1	函数	(112)
4.1.1	一个 C 语言程序例子	(112)
4.1.2	函数定义	(114)
4.1.3	函数参数和函数的值	(116)
4.1.4	函数的返回值	(117)
4.2	函数的调用	(118)
4.2.1	函数调用一般格式	(118)
4.2.2	函数调用的方式	(120)
4.2.3	函数的传值调用	(121)
4.2.4	函数的嵌套调用	(123)
4.3	一维数组	(125)
4.3.1	数组概念	(125)
4.3.2	一维数组	(127)
4.3.3	一维数组应用举例	(128)
4.3.4	数组名作函数参数的调用	(131)
4.4	二维数组	(133)
4.4.1	二维数组概念	(133)
4.4.2	二维数组引用	(134)
4.4.3	二维数组应用举例	(135)
4.5	进阶部分知识	(137)
4.5.1	函数的递归调用	(137)
4.5.2	字符数组	(141)
4.5.3	字符串	(143)
4.5.4	字符数组的应用	(145)
4.6	习题四	(146)
4.7	实训四	(151)
第 5 章	指针	(155)
5.1	几个大家熟悉的指针	(155)
5.1.1	旅馆客房指针概念	(155)
5.1.2	几个日常生活中指针概念	(156)
5.1.3	VFP 中指针概念	(156)
5.1.4	C 语言中的单元地址和单元内容	(157)
5.2	C 语言中的指针	(158)
5.2.1	指针变量	(158)

5.2.2	运算符 & 和 *	(160)
5.2.3	指针变量的引用	(162)
5.2.4	指针运算	(164)
5.2.5	指向指针的指针	(167)
5.3	指针和数组	(168)
5.3.1	一维数组的指针表示	(168)
5.3.2	二维数组的指针表示	(171)
5.3.3	指针与字符串	(174)
5.3.4	指针数组	(177)
5.4	指针和函数	(180)
5.4.1	指针作为函数的参数	(180)
5.4.2	指针作为函数的返回值	(182)
5.4.3	数组指针作为函数的参数	(184)
5.5	进阶部分知识	(187)
5.5.1	函数指针	(187)
5.5.2	main 函数中的参数	(189)
5.6	习题五	(190)
5.7	实训五	(194)
第 6 章	结构体、共用体和枚举	(199)
6.1	一个典型的实际问题	(199)
6.1.1	学生档案数据类型	(199)
6.1.2	定义结构体类型变量的方法	(200)
6.1.3	结构体变量的初始化	(202)
6.1.4	结构体变量的引用	(203)
6.2	结构体数组与函数	(204)
6.2.1	结构体数组定义	(204)
6.2.2	结构体数组的初始化	(206)
6.2.3	结构体数组的引用	(207)
6.2.4	结构体变量作函数参数	(212)
6.3	共用体数据类型	(214)
6.3.1	共用体的定义	(214)
6.3.2	共用体变量的定义	(215)
6.3.3	共用体变量的引用	(215)
6.3.4	共用体变量的应用	(218)
6.4	枚举数据类型	(219)
6.4.1	枚举的定义	(220)
6.4.2	枚举变量的定义	(220)
6.4.3	枚举变量的引用	(221)
6.5	进阶部分知识	(225)
6.5.1	指向结构体变量的指针	(225)

6.5.2 指向结构体数组的指针	(226)
6.5.3 用结构体变量指针作函数参数	(228)
6.6 习题六	(230)
6.7 实训六	(235)
第7章 文件	(238)
7.1 一个典型的实际问题	(238)
7.1.1 学生档案数据文件	(238)
7.1.2 文件概念	(240)
7.1.3 文件类型指针	(241)
7.2 文件的打开与关闭	(243)
7.2.1 文件打开	(243)
7.2.2 文件关闭	(244)
7.3 文件的顺序读写	(244)
7.3.1 输入和输出一个字符	(244)
7.3.2 输入和输出一个字符串	(246)
7.3.3 数据块读写函数	(248)
7.3.4 格式化输入输出函数	(251)
7.4 文件的随机读写	(252)
7.4.1 文件随机读写概念	(252)
7.4.2 文件的定位	(252)
7.4.3 文件的出错检测	(255)
7.4.4 文件输入输出小结	(255)
7.5 编译预处理	(256)
7.5.1 宏定义	(256)
7.5.2 带参数的宏定义	(257)
7.5.3 文件包含	(258)
7.6 进阶部分知识	(260)
7.6.1 非缓冲文件系统的特点	(260)
7.6.2 文件打开	(261)
7.6.3 文件关闭	(262)
7.6.4 文件的读写	(262)
7.6.5 缓冲区的设置	(262)
7.6.6 文件重定向	(263)
7.7 习题七	(264)
7.8 实训七	(267)
第8章 C语言综合训练	(269)
8.1 简单应用程序	(269)
8.1.1 换币问题	(269)
8.1.2 整数排序	(271)
8.1.3 约瑟夫问题	(273)

8.2 一般应用程序	(275)
8.2.1 分屏显示文件内容	(275)
8.2.2 给小学生出四则混合题目	(277)
8.2.3 创建一个文本编辑器	(280)
8.3 图形应用程序	(288)
8.3.1 画直线	(288)
8.3.2 多条直线组成的图形	(289)
8.3.3 云彩图形	(292)
8.4 C 语言课程设计	(296)
8.4.1 目的要求	(296)
8.4.2 实训时间	(297)
8.4.3 实训内容	(297)
8.4.4 参考题目	(297)
8.4.5 考核标准	(298)
附录 A Turbo C 2.0 常用错误提示解释	(299)
参考文献	(311)

第 1 章 C 语言程序设计基础

C 语言(或简称 C)是继 BASIC 语言、FORTRAN 语言、COBOL 语言和 Pascal 语言之后问世的一种通用计算机程序设计语言。随着计算机的迅速发展和广泛应用,C 语言在计算机软件开发中的作用日益重要,越来越显示出它的魅力,已成为世界上广泛流行的、最有发展前途的计算机高级语言。它适用于编写各种系统软件(著名的 FoxPro 系统软件就是用 C 语言编写成功的),也适用于编写各种应用软件,特别是当前在 Internet 上最为流行的电子商务软件,大部分程序也都是用 C 语言编写的。随着 C 语言功能的不断增强,使用不断普及,特别是在计算机上推出的各种版本的 C 语言系统和工具(例如,Turbo C,C,C++,Visual C 等版本),受到越来越多用户的青睐。

本章主要通过一个非常简单的、而且大家都很熟悉的例子,来揭开 C 语言的神秘面纱,介绍 C 语言的组成、特点、输入输出函数以及如何实现 C 语言程序的运行等内容。通过本章的学习,使学生了解 C 语言程序的结构,掌握 C 语言的结构、输入输出函数的基本格式和功能,学会如何实现 C 语言程序的执行及方法和步骤。

1.1 一个大家熟悉的例子

1.1.1 一个简单的 C 语言程序例子

用 C 语言编写的程序非常简单,请看下面的一个大家熟悉的例子。

【例题 1.1】假设已知两个正整数 m 和 n 的值,求 m 与 n 之和是多少?

要实现 m 与 n 之和,使用 C 语言可以编写以下的程序。

【程序清单】

```
main( )
{
    int m,n,o;
    scanf("%d%d",&m,&n);
    o = m + n;
    printf("%d",o);
}
```

当运行该程序后,在屏幕上出现一个闪烁的光标,要求用户从键盘上输入 m 和 n 的值,当键入如下信息:

123 118↵

其中,123 和 118 是赋给 m 和 n 的值;┐表示空格,作为 123 和 118 的分隔符;↵表示回车键。(当然,也可以输入一个 123 按一个回车键,再输入一个 118 再按一个回车键。)

屏幕上接着显示如下信息:

241

这就是 m 与 n 之和,是该程序运行后的结果。

如果要求 m 与 n 的乘积,则只要将第 5 行语句中的“ $m+n$ ”修改成“ $m*n$ ”即可,其他不需要修改任何一个语句。C 语言程序是否非常简单?

如果要使屏幕中的信息有提醒用户的功能,该怎样设计呢?请看下一个例题。

【例题 1.2】假设已知两个正整数 m 和 n 的值,求 m 与 n 之和是多少?要求屏幕输出有提醒用户的功能。

要达到以上目的,使用 C 语言可以编写以下的程序。

【程序清单】

```
main( )
{
    int m,n,o;
    printf("%40s\n","* * * * *");
    printf("%40s\n"," 求 m 和 n 之和  ");
    printf("%40s\n","* * * * *");
    printf("%40s\n","请输入 m 和 n 的值:");
    scanf("%d%d",&m,&n);
    o = m + n;
    printf("%40d%s%d%s",m,"+",n,"=");
    printf("%d",o);
}
```

当运行该程序后,在屏幕上出现如下信息:

```
* * * * *
  求 m 和 n 之和
* * * * *
请输入 m 和 n 的值:
```

这时,闪烁的光标在等待用户从键盘输入 m 和 n 的数值,当键入如下信息:

123 118

屏幕上接着显示如下信息:

123 + 118 = 241

这就是 m 与 n 之和,是该程序运行后的结果。该程序与例题 1.1 相比,屏幕输出具有直观和清楚的特点,使用户一目了然。但是其程序的核心还是与例题 1.1 相同,只是在程序的前、后使用了输出语句修饰屏幕的输出结果,使程序更加接近实际应用情况,更加实用,更加满足用户的需要。

同样,将程序稍加修改,可以将 $m+n$ 修改成求 m 的 n 次方(m^n),也可修改成求 m 的 n 阶乘次方($m^{n!}$)等。

1.1.2 程序的简单分析

为了分析方便,我们将例题 1.1 的程序增加了几个注释,请看经过注释后的例题 1.1。

【程序清单】

```

main( )                /* 第1行:主函数 */
{                      /* 第2行:主函数体的开始,用花括号表示 */
    int m,n,o;          /* 第3行:定义变量类型语句 */
    scanf("%d%d",&m,&n); /* 第4行:数据输入语句 */
    o = m + n;          /* 第5行:赋值语句 */
    printf("%d",o);     /* 第6行:数据输出语句 */
}                      /* 第7行:主函数体的结束,用花括号表示 */

```

下面就来分析该程序。

(1) 该程序作为一个文件存放在磁盘上,其扩展名为 .C(系统默认)。

(2) 程序的第1行 main()表示主函数,每一个C语言程序都必须有一个 main()函数。函数体由一对花括号{ }括起来。

(3) 程序的第3行是定义变量 m,n 和 o 类型的语句。int m,n,o 表示定义 m,n,o 这3个变量为整型类型。语句与语句之间用分号“;”分隔。

(4) 程序的第4行是数据输入语句,其中“%d”表示输入的数据类型为十进制整数类型。“&m,&n”中的“&”表示“取地址”。整个语句的作用就是,接受用户从键盘上输入的2个整数类型数值,分别输入到变量 m 和 n 的地址所标志的存储单元中。假设,从键盘上输入 123 和 118 两个整型数值,存储单元可以看成是一个盒子,盒子的名称分别为 m 和 n,盒子中分别存放 123 和 118 两个数值。

(5) 程序的第5行是赋值语句,其中“=”表示赋值号,赋值号的左边是变量 o,右边是表达式 m + n。语句的作用就是将右边表达式的值赋给左边的变量 o,也就是执行了赋值语句后,在 o 变量的存储单元中,存放的数值为 241,该数值的类型也是整型类型。

(6) 程序的第6行是输出语句,其中“%d”表示输出数值的类型为十进制整数类型,该语句的作用是输出变量 o 的值。

(7) 程序中的“/* */”表示注释符号,在注释符号中间是注释内容,该内容可以由任何的字符构成,系统不执行注释内容。注释的作用是给程序员阅读程序带来方便。而且程序中做注释(文档),这是优秀程序员必备的良好习惯。

1.1.3 C 语言程序结构

通过以上的分析,可以看出 C 语言程序的基本结构。

1. C 语言程序是由函数构成的

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 main()主函数,也可以包含一个主函数和若干个其他函数。例题 1.1 源程序是由一个主函数构成的。下面再来看一个程序,该程序包含 2 个函数,一个是主函数,另一个是自定义函数。

【例题 1.3】假设已知 2 个正整数 m 和 n 的值,比较它们的大小,并输出其中的最大值。要实现这一功能,使用 C 语言可以编写以下的程序。

【程序清单】

```

main( )                /* 主函数 */
{
    int m,n,o;          /* 定义变量 */

```

```

scanf("%d%d",&m,&n);    /* 输入变量 m 和 n 的值 */
o = max(m,n);           /* 调用 max 函数,并将得到的值赋给变量 o */
printf("max = %d",o);    /* 输出 o 的值 */
scanf("%d",&m);
}

int max(x,y)             /* 定义 max 函数,函数值为整型,x,y 为形式参数 */
int x,y;                 /* 对形参 x,y 作类型定义 */
{
    int z;               /* 定义变量 z 类型 */
    if (x > y) z = x;      /* 比较 x,y 的大小,如果 x 大则赋给变量 z */
    else z = y;           /* 否则 y 赋给变量 z */
    return(z);            /* 返回 z 的值,通过 max 带回调用处 */
}

```

当运行该程序并输入 m 和 n 数值时,假设分别输入 123 和 118 后,在屏幕上输出以下信息:

max = 123

从以上程序来看,该程序是由两个函数构成,一个是 main() 主函数,另一个是 max(x,y) 自定义函数。在主函数中调用自定义函数,并将实形参数 m 和 n 的值分别传递给 x 和 y,并执行比较,结果最大值是 123 并将该值返回去,程序结果是 max = 123。

综上所述可知,函数是 C 程序的基本单位。被调用的函数可以是系统提供的库函数(例如 printf 和 scanf 函数),也可以是用户根据需要自己定义的编制设计的函数(例如, max 函数)。C 程序的函数相当于其他计算机语言中的子程序,用函数来实现特定的功能。

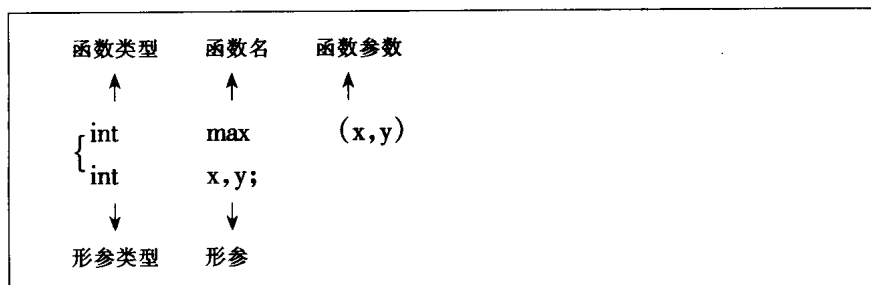
可以说,C 是函数式的语言。程序全部工作都是由函数来完成的。C 的函数库十分丰富,标准 C 提供了一百多个库函数,Turbo C 和 MS C 4.0 提供了三百多个库函数,Visual C 提供的函数库就更多了。

2. 函数的格式

一个函数由 2 部分组成:说明部分和函数体。

(1) 函数的说明部分。主要包括函数名、函数类型、函数属性、函数参数(形参)名、形式参数类型等。一个函数名后面必须跟一对圆括号,函数参数可以没有,例如,main()。

例如,例题 1.3 中的 max 函数的说明部分为:



(2) 函数体。即函数说明部分下面的花括号“{ }”内的部分。如果一个函数内有多多个花括号,则最外层的一对“{ }”为函数体的范围。

函数体一般包括函数的变量定义和执行部分 2 部分内容:

- ① 变量定义。例如在例题 1.3 中 main 函数中的“int m,n,o;”。
- ② 执行部分。由若干个语句构成,每个语句之间用“;”分隔。

3. C 语言程序的执行

一个 C 语言程序总是从 main() 主函数开始执行的,而不论 main() 函数在整个程序中的位置。一般来说,main() 函数可以放在程序的最前面,也可以放在程序的最后面,或在一些函数之前在另一些函数之后。

4. C 语言程序的书写格式

C 语言本身对书写格式要求很宽松,所以,它的书写格式非常自由。但是,由于 C 语言语句比较简洁精练,它的易读性较差,这就要求在书写格式上按照一定规格,增加其易读性。C 语言的书写格式,读者可以从后面的程序中逐渐学会,在这里我们仅做一般性的简单介绍。

(1) C 语言中一行内可以写一个语句,也可以写几个语句。一个语句可以在一行内完成,也可以分成几行完成,C 语言程序中没有行号。

(2) C 语言中花括号“{ }”用的比较多,一般情况下,左右花括号各占一行,并且需上下对齐,这样便于检查花括号的成对性。

(3) 在 C 语言程序中运用缩格写法对增加易读性十分重要,有时由于缩格运用的不好,反而将程序理解错了,缩格写法需在实践中学习。

例如,一个函数的函数体要比定义函数的花括号缩 2 格;一个循环语句的循环体或条件语句的 if 体、else 体等都要做缩格处理。这样做有利于程序的阅读和查错,特别是在编写实际应用的程序时尤为重要。

(4) 一般 C 语言程序使用小写字母来书写程序,特别是函数名必须使用小写字母来表示,C 语言程序中的大写字母一般表示常量。

5. C 语言没有输入输出语句

C 语言本身没有输入和输出语句,输入和输出的操作是由库函数 scanf 和 printf 等函数来完成的,所以,scanf 和 printf 本身的格式就是函数的格式。

综上所述,不难看出一个 C 语言编写的程序的元素及其之间关系:

程序 → 文件 → 函数 → 语句 → 单词 → 字符

对上述结构可以解释和理解如下:

- ① 一个 C 语言程序是由一个或多个文件组成的;
- ② 一个文件又由一个或多个函数组成;
- ③ 函数是组成程序的基本单位;
- ④ 一个函数又是由若干个语句构成函数体的;
- ⑤ 一个语句是由若干个单词组成的;
- ⑥ 单词是由字符组成的;
- ⑦ 字符是组成 C 语言程序的最小元素。

1.1.4 字符及字符集

1. 字符

组成 C 语言源程序中的最小元素是字符, C 语言程序中的字符序列必须是符合该语言语法规则的字符串。因此, 要求 C 语言程序中每个字符都是合法的字符。

(1) 字母和数字

小写字母: a b c d e f g h i j k l m n o p q r s t u v w x y z 等。

大写字母: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 等。

数字符: 0 1 2 3 4 5 6 7 8 9

(2) 空白字符

空白字符是由空格符、回车符、换行符、水平制表符等字符组成的。

(3) 空字符

空字符是指 ASCII 码值为零的那个字符。该字符在 C 语言中有特殊的作用, 可用来作为字符串的结束符。

(4) 标点和特殊字符

这类字符见表 1-1 所示。

表 1-1 C 字符集中的标点和特殊字符一览表

字符	名称	字符	名称	字符	名称
,	逗号	{	左花括号	#	数字号
.	点	}	右花括号	%	百分号
;	分号	<	小于号	&	脱字号
:	冒号	>	大于号	^	和号
'	单引号	!	惊叹号	*	乘号
"	双引号		竖线	-	减号
(	左括号	/	斜线	=	等号
)	右括号	\	反斜线	+	加号
[	左方括号	~	非		
]	右方括号	_	下划线		

2. 单词

单词是由若干个有序的字符组成的, 单词的集合称为词汇。C 语言的词汇是相当丰富的, 通常可将其分为: 标识符、关键字、运算符、分隔符、常量、字符串、注释符等 7 种类型。

(1) 标识符

① 标识符的组成: 标识符是由字母, 下划线和数字组成的字符序列。要求第一个字符必须是字母或下划线。

② 标识符的功能: 主要用来给程序中所使用的变量、函数、语句标号、类型定义等起名字之用。

③ 使用时注意事项: a). 标识符对所用的字符个数不限制, 但在具体使用时受机器本身的限制, 有些只能识别标识符的前 8 个字符; b). 多数 C 编译系统对大小写字母是区分的; c).

要求标识符名字起的尽量有意义,主要是为了便于阅读;d).要注意标识符的合法性。例如:

下面的标识符是合法的:

a100, x_add, int_cnt, _form, Ax, x_y_o 等。

下面的标识符是不合法的:

26d, #mon, -cnt, ab*c, m.p, \$5d 等。

(2) 关键字

① 关键字的组成:关键字是一种具有特定含义的预先已定义的标识符。关键字也称为保留字。

② 关键字的功能:主要用来给系统作特定的功能之用,例如, int 是用来作标识整形型数据之用。

③ 下面介绍几种特定的关键字:

a. 标识类型的关键字: int, char, float, double, long, unsigned, struct, union, enum, auto, extern, static, register, typedef, void 等。

b. 标识控制流的关键字: goto, return, break, continue, if, else, do, while, switch, case, default 等。

c. 标识预处理的关键字: define, include, undef, ifdef, ifndef, endif, line 等。

d. 其他关键字: sizeof, asm, fortran, ada, pascal 等。

(3) 运算符

① 运算符的组成:运算符是用来表示某种运算的符号,多数运算符是由一个字符组成的,也有的运算符是由多个字符组成。

② 运算符的功能:主要用来进行各种算术运算、关系运算、字符运算、逻辑运算等。在以后的各章节中会详细介绍有关运算符的功能。

(4) 分隔符

① 分隔符的组成:分隔符是用来分隔变量、数据、表达式等多个单词的符号。一般使用逗号和空白符作为分隔的符号。

② 运算符的功能:主要用来分隔参数表中的各个参数,这些参数可以是变量、常量、表达式等。

(5) 常量

常量的组成:常量是一种在程序中其值不会改变的量。C 语言中常量由数字常量和字符常量两类组成。

① 数字常量又分为整型数常量和浮点数常量。

② 字符常量又分为字符常量和字符串常量。

(6) 字符串

字符串是指用双引号(" ")的一串字符。例如,下面都是合法的字符串:

"abc"、"我是大学生"、"21345"、"xx \ yy \ zz"等。

(7) 注释符

注释符是用来标识注释或提示信息的。程序中的注释符是不被编译也不被执行的,其用途在于增加程序的可读性,起到说明和备忘的作用。C 语言的注释符是以“/*”开始的并以“*/”结尾的一对注释符构成的。