

高 职 高 专 计 算 机 系 列 规 划 教 材



C语言程序设计 (第2版)

陈孟建 邹玉金 熊传光 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

高职高专计算机系列规划教材

C 语言程序设计

(第2版)

陈孟建 邹玉金 熊传光 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、课后练习和实训三部分内容组成。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,特别适合职业技术学院学生动手能力的培养。

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、C语言程序设计的常用算法、指针、结构体、共用体和枚举、C语言程序综合实训等内容。

本书可作为高职高专的学生学习C语言程序设计的教材或参考用书,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, 邹玉金, 熊传光编著. 2版. —北京: 电子工业出版社, 2006.6

(高职高专计算机系列规划教材)

ISBN 7-121-02635-X

I. C… II. ①陈… ②邹… ③熊… 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(2006)第049336号

责任编辑: 吕 迈 (Lumai@phei.com.cn)

印 刷: 涿州市京南印刷厂

出版发行: 电子工业出版社

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: 518千字

印 次: 2006年6月第1次印刷

印 数: 5000册 定价: 27.00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) 88254043。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

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、课后练习和实验内容三部分组成。采用循序渐进的内容安排、通俗易懂的讲解方法,书中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,特别适合职业技术学院学生动手能力的培养。

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、实例主线。其中, C 语言的学科主线是作者编写的主线,实例主线是学生学习的主线。

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,并强调实例教学、实训教学。

3. 本书在编写结构中,采用先提出问题,再进行问题分析,然后给出解决问题的方法,最后通过实例举一反三,使学生增强分析问题和解决问题的能力。同时,每章有强调技能提高的内容,如进阶知识部分。

4. 本书所有的例题均可在 Turbo C 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学院 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,得到了陈惠珍、沈美莉、刘逸平、张寅利、陈奕婷、田文雅、商玮等专家、教授们的帮助,在此,表示衷心地感谢!

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

第2版前言

本书自2002年5月发行第1版以来，至今已有四年的时间。在这期间，收到了不少读者的来信，对本书提出了许多宝贵的意见和建议，在此深表谢意。

近几年我们经过教学实践和社会调查，对原有版本的内容进行了适当的修正和补充。从对教材科学性、理论性和先进性的考虑，我们对教材做了较大的删除、新增和修改。特别是新增了“C语言程序设计的常用算法”一章。在这一章里我们收集了许多典型的例子，包括累加、累乘、整除、二查、三排、三遍历算法等，这对学生掌握计算机解题的基本思想、方法和步骤，学好C语言有很大的帮助。

全书内容包括：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、C语言程序设计的常用算法、指针、结构体、共用体和枚举、C语言程序综合实训等。在本书的编写手法上，从培养和提高学生的计算机应用能力着手，叙述深入浅出。本书可读性、可操作性强，既方便老师教，又方便学生学，可满足不同基础的学生的需要。

本书在编写过程中，得到了沈美莉、陈惠玲、刘逸平、张寅利、陈奕婷、田文雅、商玮、盘宏华等专家、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

2006年1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	(3)
1.1.3 C语言程序的组成要素	(3)
1.1.4 字符及字符集	(5)
1.2 C语言的由来、特点和发展	(7)
1.2.1 C语言的由来	(7)
1.2.2 C语言的特点	(8)
1.2.3 C语言的发展	(9)
1.3 输入与输出函数	(9)
1.3.1 标准格式输入函数 scanf()	(9)
1.3.2 标准格式输出函数 printf()	(11)
1.3.3 单个字符输入函数 getchar()	(16)
1.3.4 单个字符输出函数 putchar()	(17)
1.4 C语言程序的实现	(18)
1.4.1 C语言程序实现的方框图	(18)
1.4.2 C语言的编辑	(19)
1.4.3 编译与运行程序	(22)
1.4.4 C语言程序的错误分类	(24)
1.5 进阶部分知识	(26)
1.5.1 转义字符	(26)
1.5.2 scanf函数格式说明符	(26)
1.5.3 printf函数格式说明符	(27)
习题1	(32)
实训1	(33)
第2章 数据描述与基本操作	(36)
2.1 数据类型	(36)
2.1.1 一个C语言程序例子	(36)
2.1.2 数据类型	(39)
2.1.3 数据的存储格式、范围与有效位	(40)
2.1.4 赋值语句	(42)
2.2 常量与变量	(43)
2.2.1 常量	(43)
2.2.2 变量	(46)
2.2.3 变量说明语句	(47)
2.2.4 各种数值型数据间的混合运算	(47)
2.3 运算符与表达式	(48)

2.3.1 C 语言的运算符	(48)
2.3.2 算术运算符	(49)
2.3.3 算术运算符的优先级	(51)
2.3.4 算术表达式	(52)
2.4 关系运算、逻辑运算与条件运算	(54)
2.4.1 关系运算及表达式	(54)
2.4.2 逻辑运算及表达式	(55)
2.4.3 条件运算	(58)
2.5 进阶部分知识	(59)
2.5.1 赋值表达式	(59)
2.5.2 赋值中的类型转换	(60)
2.5.3 变量的初始化	(61)
2.5.4 逗号运算符	(62)
习题 2	(63)
实训 2	(65)
第 3 章 结构化程序设计	(68)
3.1 结构化程序设计思想	(68)
3.1.1 一个 C 语言程序例子	(68)
3.1.2 结构化程序设计的由来	(71)
3.1.3 结构化程序分析方法	(72)
3.1.4 结构化系统分析方法的优点	(73)
3.2 选择结构程序设计	(73)
3.2.1 简单条件分支语句	(73)
3.2.2 选择条件分支语句	(76)
3.2.3 无条件转向语句	(78)
3.3 循环结构程序设计	(80)
3.3.1 循环结构程序的概念	(80)
3.3.2 while 循环结构	(82)
3.3.3 do-while 循环结构	(83)
3.3.4 for 循环结构	(84)
3.3.5 双重循环结构	(87)
3.3.6 break、continue 及 exit 语句	(89)
3.3.7 几种循环的比较	(92)
3.4 进阶部分知识	(93)
3.4.1 条件分支语句的嵌套	(93)
3.4.2 开关语句	(95)
3.4.3 循环语句的嵌套	(98)
习题 3	(101)
实训 3	(103)
第 4 章 模块化程序设计	(107)

4.1	函数	(107)
4.1.1	一个C语言程序例子	(107)
4.1.2	函数的定义	(109)
4.1.3	函数参数和函数的值	(111)
4.1.4	函数的返回值	(112)
4.2	函数的调用	(113)
4.2.1	函数调用的一般格式	(113)
4.2.2	函数调用的方式	(115)
4.2.3	函数的传值调用	(115)
4.2.4	函数的嵌套调用	(118)
4.3	一维数组	(120)
4.3.1	数组概念	(120)
4.3.2	一维数组	(122)
4.3.3	一维数组应用举例	(123)
4.3.4	数组名用做函数参数的调用	(126)
4.4	二维数组	(128)
4.4.1	二维数组的概念	(128)
4.4.2	二维数组的引用	(129)
4.4.3	二维数组的应用举例	(130)
4.5	进阶部分知识	(132)
4.5.1	函数的递归调用	(132)
4.5.2	字符数组	(135)
4.5.3	字符串	(137)
4.5.4	字符数组的应用	(139)
	习题4	(141)
	实训4	(145)
第5章	C语言程序设计的常用算法	(149)
5.1	累加、累乘、整除性算法	(149)
5.1.1	算法的基本概念	(149)
5.1.2	累加、累乘算法	(149)
5.1.3	整除性算法	(151)
5.2	最大、最小值和双重循环算法	(156)
5.2.1	最大值和最小值算法	(156)
5.2.2	验证哥德巴赫猜想	(159)
5.2.3	矩阵(二维数组)算法	(161)
5.2.4	杨辉三角形算法	(165)
5.3	二查、三排、三遍历算法	(168)
5.3.1	二种查找算法	(168)
5.3.2	三种排序算法	(170)
5.3.3	三遍历算法	(173)

习题 5	(178)
实训 5	(181)
第 6 章 指针	(188)
6.1 几个大家熟悉的指针	(188)
6.1.1 旅馆客房的指针概念	(188)
6.1.2 几个日常生活中的指针概念	(189)
6.1.3 VFP 中的指针概念	(189)
6.1.4 C 语言中的单元地址和单元内容	(190)
6.2 C 语言中的指针	(191)
6.2.1 指针变量	(191)
6.2.2 运算符 & 和 *	(193)
6.2.3 指针变量的引用	(195)
6.2.4 指针运算	(197)
6.2.5 指向指针的指针	(200)
6.3 指针和数组	(201)
6.3.1 一维数组的指针表示	(201)
6.3.2 二维数组的指针表示	(204)
6.3.3 指针与字符串	(206)
6.3.4 指针数组	(209)
6.4 编译预处理	(212)
6.4.1 宏定义	(212)
6.4.2 带参数的宏定义	(213)
6.4.3 文件包含	(214)
6.5 文件操作	(216)
6.5.1 文件的概念	(216)
6.5.2 文件的打开与关闭	(218)
6.5.3 文件的顺序读、写	(219)
6.5.4 数据块读、写函数	(220)
6.5.5 文件的随机读、写	(221)
6.6 进阶部分知识	(223)
6.6.1 函数指针	(223)
6.6.2 main 函数中的参数	(226)
习题 6	(227)
实训 6	(230)
第 7 章 结构体、共用体和枚举	(235)
7.1 一个典型的实际问题	(235)
7.1.1 学生档案的数据类型	(235)
7.1.2 定义结构体类型变量的方法	(236)
7.1.3 结构体变量的初始化	(238)
7.1.4 结构体变量的引用	(239)

7.2 结构体数组与函数	(240)
7.2.1 结构体数组的定义	(240)
7.2.2 结构体数组的初始化	(242)
7.2.3 结构体数组的引用	(242)
7.2.4 结构体变量用做函数参数	(247)
7.3 共用体数据类型	(249)
7.3.1 共用体的定义	(250)
7.3.2 共用体变量的定义	(251)
7.3.3 共用体变量的引用	(252)
7.3.4 共用体变量的应用	(253)
7.4 枚举数据类型	(254)
7.4.1 枚举的定义	(254)
7.4.2 枚举变量的定义	(255)
7.4.3 枚举变量的引用	(256)
7.5 进阶部分知识	(260)
7.5.1 指向结构体变量的指针	(260)
7.5.2 指向结构体数组的指针	(261)
7.5.3 用结构体变量指针用做函数参数	(262)
习题 7	(264)
实训 7	(269)
第 8 章 C 语言程序综合实训	(272)
8.1 简单应用程序	(272)
8.1.1 换币问题	(272)
8.1.2 整数排序	(274)
8.1.3 约瑟夫问题	(276)
8.2 一般应用程序	(277)
8.2.1 分屏显示文件的内容	(277)
8.2.2 给小学生出四则混合运算题目	(280)
8.2.3 创建一个文本编辑器	(283)
8.3 学生成绩管理应用程序	(290)
8.3.1 功能模块与主界面	(290)
8.3.2 各子菜单项的操作	(291)
8.3.3 源程序清单	(293)
8.4 C 语言课程设计	(303)
8.4.1 目的要求	(303)
8.4.2 实训时间	(303)
8.4.3 实训内容	(303)
8.4.4 参考题目	(303)
8.4.5 考核标准	(304)
附录 A Turbo C 2.0 常用错误提示解释	(305)
参考文献	(312)

第1章 C语言程序设计基础

C语言是继 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。随着电子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。随着 C 语言功能的不断增强，应用不断普及，特别是在微机上推出的各种版本的 C 语言系统和工具（例如，Turbo C、Borland 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，要求：

1. 了解 C 语言程序的结构；
2. 掌握组成 C 语言程序的最小元素（字符和字符集）；
3. 掌握输入/输出函数的基本格式和功能；
4. 学会如何实现 C 语言程序的运行、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 1.1】 假设已知两个正整数 m 和 n 的值，求 m 与 n 之和。

【程序清单】

```
#include "stdio.h"
void 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，输入如下信息：

123_118 ✓

其中, 123 和 118 是要赋给 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 按一下回车键, 再输入一个 118 按一下回车键)。

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, 则只要将第五行语句中的 $m+n$ 修改成 $m*n$ 即可, 其他不需要修改。

如果要使屏幕中的信息有提醒用户的功能, 请看下一个例题。

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

【程序清单】

```
#include "stdio.h"
void 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);
}
```

运行结果:

```
*****
      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, 当输入如下信息:

123_118 ↵

屏幕上接着显示信息:

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 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。但是该程序的核心思想还是与例 1.1 相同, 只是在程序中使用了输出语句用来修饰屏幕的输出结果, 使程序更加实用、更加满足用户的需要。

1.1.2 程序的简单分析

为了分析方便,例 1.1 增加了几行注释,请看经过注释后的例 1.1。

【程序清单】

```
#include "stdio.h"      /* 第一行: 输入/输出函数编译预处理命令 */
void main()             /* 第二行: 主函数 */
{                       /* 第三行: 主函数体的开始, 用花括号 "{" 表示 */
    int m,n,o;           /* 第四行: 定义变量类型语句 */
    scanf("%d%d", &m, &n); /* 第五行: 数据输入语句 */
    o=m+n;              /* 第六行: 赋值语句 */
    printf("%d",o);      /* 第七行: 数据输出语句 */
}                       /* 第八行: 主函数体的结束, 用花括号 "}" 表示 */
```

分析:

(1) 该程序作为一个文件存放在磁盘上, 扩展名为.c (系统默认)。

(2) 程序的第一行是输入/输出函数编译预处理命令。

(3) 程序的第二行 `main()` 表示主函数, 每一个 C 语言程序都必须有一个 `main()` 函数。函数体是由一对花括号 “{}” 括起来的。

(4) 程序的第四行是定义变量 `m,n` 和 `o` 类型的语句。`int m,n,o` 表示定义 `m,n,o` 三个变量为整型类型。语句与语句之间用分号 “;” 分隔。

(5) 程序的第五行是数据输入语句, 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两个整数类型数值, 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。

(6) 程序的第六行是赋值语句, 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, 在变量 `o` 的存储单元中, 存放的数值为 241, 该数值的类型也是整数类型。

(7) 程序的第七行是输出语句, 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。

(8) 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。

1.1.3 C 语言程序的组成要素

通过以上的分析, 可以看出 C 语言程序的基本组成结构。

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主函数 `main()`, 也可以是包含一个主函数和若干个其他函数。例 1.1 源程序是由一个主函数构成的。下面再来看例 1.3, 该程序包含两个函数, 一个是主函数, 另一个是自定义函数。

【例 1.3】 假设已知两个正整数 m 和 n 的值, 比较它们的大小, 并输出其中的最大值。

【程序清单】

```
#include "stdio.h"          /* 输入/输出函数编译预处理命令 */
void 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);             /* 调用函数 max, 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)                /* 定义函数 max, 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 返回到调用处 */
}
```

当运行程序并输入 m 和 n 的值, 假设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的, 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, 并将实在参数 m 和 n 的值分别传递给形式参数 x 和 y , 并进行比较, 结果最大值是 123, 同时将该值返回到函数调用处, 最后, 程序结果是 `max=123`。

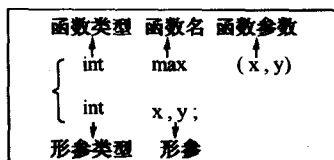
综上所述可知, 函数是 C 语言程序的基本单位, 被调用的函数可以是系统提供的库函数 (例如, `printf` 和 `scanf` 函数), 也可以是用户自己定义设计的函数, 如函数 `max`。C 语言程序的函数相当于其他计算机语言中的子程序, 用函数来实现特定的功能。

可以说, C 语言是函数式的语言, 程序的全部工作都是由函数来完成的。C 语言的库函数十分丰富, 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 提供的库函数就更多了。

2. 函数的构成

一个函数由以下两部分组成。

(1) 函数的说明部分。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。例如, 例 1.3 中的函数 `max` 的说明部分为:



一个函数名后面必须跟一对圆括号, 函数参数可以默认, 如主函数 `main()`。

(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个花括号, 则最外层的一对花括号为函数体的范围。

函数体一般包括以下两部分内容:

① 变量定义。例如, 在例 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。

一个 C 语言程序总是从主函数 main () 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。

3. 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, C 语言程序的书写格式非常自由。但是, 由于 C 语言语句比较简洁精练, 所以 C 语言程序的易读性较差, 这就要求在书写格式上按照一定规格, 增加易读性。对于 C 语言的书写格式, 读者可以从后面的程序中逐渐学会, 在这里仅进行一般性的简单介绍。

(1) C 语言中一行内可以写一条语句, 也可以写多条语句。一条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“{ }”用得比较多, 一般情况下, 左右花括号各占一行, 并且需要上下对齐, 这样便于检查花括号的成对性。

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

(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, C 语言程序中的大写字母一般表示常量。

4. 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, 而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 语言程序是由以下构成的:

程序→文件→函数→语句→单词→字符。也就是说:

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的;

⑤ 一个语句由若干个单词组成;

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。

(1) 字母和数字。

小写字母: `abcdefghijklmnopqrstuvwxyz`

大写字母: `ABCDEFGHIJKLMNOPQRSTUVWXYZ`

数字符: `0123456789`

(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。这类字符如表 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	!	感叹号	*	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	_	下画线		

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 语言的词汇相当丰富, 通常可分为: 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、下画线和数字组成的字符序列, 其中第一个字符必须是字母或下画线。

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:

- 标识符对所用的字符个数不受限制, 但在具体使用时受机器本身限制, 有些只能识别标识符的前 8 个字符;

- 多数 C 语言编译系统对大小写字母是区分的;

- 要求标识符名字起的尽量有意义, 主要是为了便于阅读;

- 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_`

下面的标识符是不合法的:

26d、#mon、-cnt、ab*c、m.p、\$5d

(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: 主要用来实现系统特定的功能。例如, int 用做标识整型数据。

③ 下面介绍几种特定的关键字:

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

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

- 标识预处理的关键字: define、include、undef、ifdef、ifndef、endif、line 等。

- 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 分隔符的功能: 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。常量是一种在程序中值不会改变的量。C 语言中常量由数字常量和字符常量组成。

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

字符串是指用一对双引号 (" ") 括起来的一串字符。例如, 下面都是合法的字符串:

"abc"、"我是大学生"、"21345"、"xx\yy\zz"

(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 "*/" 结束。

1.2 C 语言的由来、特点和发展

1.2.1 C 语言的由来

自从 1946 年世界上出现的第一台电子计算机以来, 随着在应用领域的不断扩展普及,