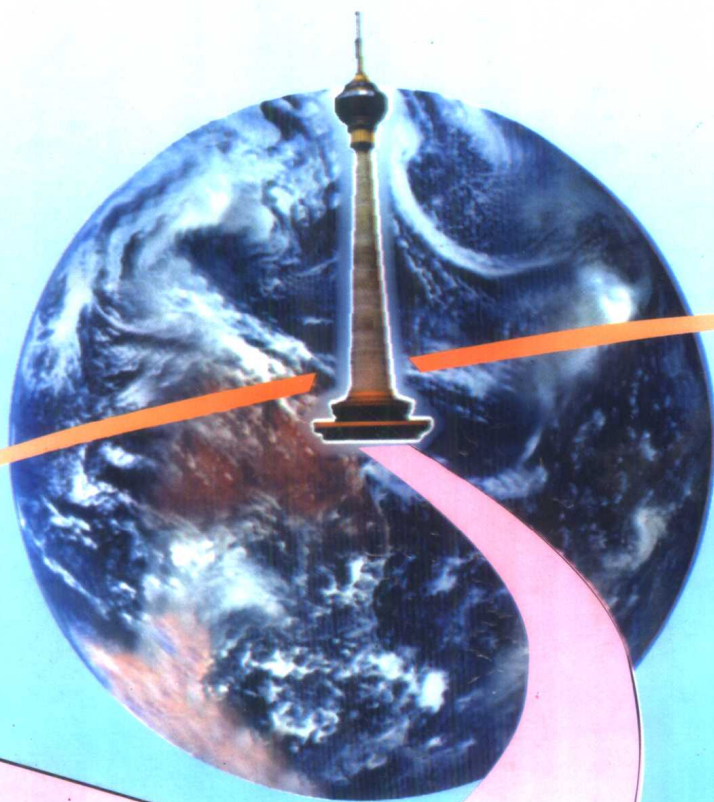




全国非计算机专业等级考试、
自学考试辅导丛书

新编初级C语言程序设计 自学辅导

本丛书编委会



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

72.8742

C421

72.8742

S-C421

阅 8 清

3-13-1
全国非计算机专业等级考试、自学考试辅导丛书

· 第三册 ·

新编初级 C 语言程序设计

自学辅导

本丛书编委会

电子工业出版社

内 容 提 要

本书是全国非计算机专业等级考试、自学考试辅导丛书的第三册，全书以通俗、浅显的文字介绍了C语言的基础知识和程序设计技术。全书突出基本技能的培养，并配有作业及答案。书中收集了初学者易出的错误，并给予答疑辅导。本书适合于参加各类计算机等级考试的读者自学使用，亦可作为计算机基础教育的入门教材。

计算机专业等级考试、自学考试辅导丛书
新编教材 C 语言程序设计自学辅导

本丛书编委会
责任编辑：应月燕 杨丽娟

电子工业出版社出版
北京市海淀区万寿路 173 信箱(100036)
电子工业出版社发行 各地新华书店经销
中国电影出版社印刷厂印刷

*

开本 787×1092 毫米 1/16 印张 14.25 字数 344 千字

1996 年 1 月第一版 1996 年 1 月北京第一次印刷

印数 8000 册 定价：17.00 元

ISBN 7-5053-3387-9/TP·1307

全国非计算机专业等级考试、自学考试辅导丛书编委会

顾 问 刘乃琦

策 划 王明君

编 委 许 远 吴 跃 刘力舰 李培茂

前 言

随着科学技术的迅猛发展,计算机已成为各个学科领域不可缺少的应用工具,计算机知识和应用能力已成为当代大学生知识和能力结构的一个重要组成部分,也是我国教育培养跨世纪人才最突出的需要加强的环节之一。目前在高校中普遍开展的计算机知识和应用能力等级考试正有效地推动这一目标的实现。1993年12月国家教委考试中心颁布的在全国进行计算机应用能力认证考试文件,这必将进一步推动全社会学习计算机、使用计算机的热潮。与此有关的教材和参考资料的需求与日俱增。

到目前为止,各省、自治区、直辖市都举办了计算机等级考试,此类的书为数不少,本书的出版正是在充分吸收先期出版的同类图书的优点、克服存在的弊病的基础上推陈出新,更上一层楼。

为了达到这个目的,我们在编撰过程中特别注意了以下问题。

(1)适当放低起点,但不降低总的要求,充分贯彻《国家教育委员会全国计算机等级考试大纲》的要求,兼顾目前存在的多种等级考试的要求,循序渐进,深入浅出,对基本内容讲透、讲够,对于易出的错误,给予明确指出,同时对《大纲》进行适当扩充,以保证该书具有一定的实用性与超前性。

(2)注意考试科目的基本知识讲授,在掌握基础知识的同时,适当地进行基本技能的训练,而不以“习题”和“模拟试题”为主。我们认为扎扎实实地掌握基本技能,完全可以达到有关考试的要求。

(3)本书适合于读者自学,也适合于有关专业进行课堂教学,每道习题均有答案,习题本着精辟、典型的原则进行收录。

(4)本书本着实用、广谱的原则,兼容各级各类考试的要求,适合于以下几种考试:

- 各省、市、自治区组织的非计算机专业计算机等级考试
- 国家教委考试中心的非计算机专业计算机能力考试
- 国家教委考试中心的计算机专业的计算机水平和资格考试

全国非计算机专业等级考试、自学考试辅导丛书编委会

07/12/02

本书阅读方法

◁必学/了解

这部分内容是基础性的和稍具理论性的,了解它们是为了学习后续内容的需要。

◁必学/重点

这部分内容是实质性的,要求准确地理解,熟练地掌握,读者阅读时可多加推敲。

◁自学答疑

这部分的内容收集了初学者易产生的一些概念错误、操作错,它是本书的画龙点睛之处,对于缩短学习周期,提高学习效率有重要指导意义,应认真对待。

◁选 学

这部分的内容超出了考试大纲的要求,若读者在实际应用中遇到这些问题或有空余的时间,不妨一看。

◁编程技巧

这部分是一些极具实用性的技巧,当你“山穷水尽疑无路”时,它也许能让你有“柳暗花明又一村”的感觉。

◁作 业

这部分收集了一些典型的习题,要求认真完成。

◁答 案

本部分是“作业”部分的答案,也许你的答案和书中的答案不一样,这很正常,也许你的答案更先进(向你祝贺),也许我们的答案有误(欢迎指正)。

目 录

第一章 C语言概述	(1)
§ 1.1 C语言的特点	(1)
§ 1.2 C语言总体印象	(3)
§ 1.3 几个基本概念	(11)
§ 1.4 C程序开发过程	(13)
第二章 变量与常量	(16)
§ 2.1 变量	(16)
§ 2.2 整型变量	(17)
§ 2.3 实型变量	(20)
§ 2.4 字符变量	(25)
§ 2.5 常量	(27)
§ 2.6 无名常量与符号常量	(29)
第三章 表达式与赋值语句	(34)
§ 3.1 表达式及其类型	(34)
§ 3.2 赋值语句	(40)
§ 3.3 疑难分析与应用实例	(43)
第四章 输入输出控制	(50)
§ 4.1 printf 函数	(50)
§ 4.2 其它输出函数	(62)
§ 4.3 scanf 函数	(64)
§ 4.4 其它输入函数	(66)
§ 4.5 应用举例	(69)
第五章 程序的分支	(75)
§ 5.1 关系运算	(75)
§ 5.2 逻辑运算	(77)
§ 5.3 程序的分支	(80)
§ 5.4 选择语句	(89)
§ 5.5 goto 语句	(97)

第六章 循环控制	(104)
§ 6.1 计数循环(for 语句).....	(104)
§ 6.2 for 循环的嵌套.....	(110)
§ 6.3 while 循环语句	(116)
§ 6.4 do~while 语句	(122)
§ 6.5 其它循环嵌套	(125)
§ 6.6 循环编程技巧	(127)
§ 6.7 循环辅助控制	(130)
第七章 数组的初步认识	(139)
§ 7.1 数组的定义	(139)
§ 7.2 初始化与赋值	(144)
§ 7.3 一维数组的应用	(146)
§ 7.4 二维数组	(153)
§ 7.5 二维数组的应用	(161)
§ 7.6 数组的存储	(169)
第八章 函数的初步认识	(174)
§ 8.1 函数的概念	(174)
§ 8.2 全局变量与局部变量	(178)
§ 8.3 参数传递	(180)
§ 8.4 函数返回值	(186)
§ 8.5 函数的应用	(191)
§ 8.6 递归函数	(201)
§ 8.7 函数与结构化程序设计	(206)
§ 8.8 提高与技巧	(208)

第一章 C语言概述

【导读提要】本章介绍C语言的最基本格式,主要有C语言的发展史,C语言的特点,尤其要注意C语言的书写格式是怎样的。接下来介绍了C语言程序的开发过程和C语言中的一些常用概念,本章是后续章节的基础。

本章中要求熟练掌握的内容有:

- C语言的书写格式
- C语言的字符集
- C语言的关键字
- C语言的标识符
- C语言的注解风格

本章中要求了解的内容有:

- C语言的发展史
- C语言的特点
- C语言程序的开发过程

§ 1.1 C语言的特点

一、C语言发展简史

◁必学/了解

C语言是一种编译型程序设计语言,它的前身是马丁·理查德(Martin Richards)在60年代开发的BCPL语言;1970年,肯·苏姆普逊(Ken·Thomppson)在软件开发工作中,继承和发展了BCPL语言的特点,并提出了“B语言”;此后,在美国贝尔研究所进行的更新型的小型机PDP-11上的UNIX操作系统的开发工作中,戴尼斯·M·利奇(Dennis·M·Ritchie)和布朗·W·卡尼汉(Brian·W·Kernighan)对B语言做了进一步的充实和完善,于1972年推出了一种新型的程序语言——C语言。

自从C语言问世以来,就表现出较强的生命力,它从最初的UNIX操作系统的记述语言开始迅速发展成为广泛应用的通用程序设计语言;从过去仅在小型机领域中使用的C语言,成长为微型机世界中的主要程序语言之一。

二、C语言的特点

◁必学/了解

C语言有以下几个基本特点:

(1)C语言是处于汇编语言和高级语言之间的一种过渡型程序语言。C语言在程序设计语言中所处的地位,如图1-1所示。从中可以看出,C语言是比较靠近硬件与系统的,即与汇编语言比较接近。C语言既有面向硬件和系统,像汇编语言那样可以直接访问硬件的功能,

又有高级语言面向用户、容易记忆、便于阅读和书写的优点。

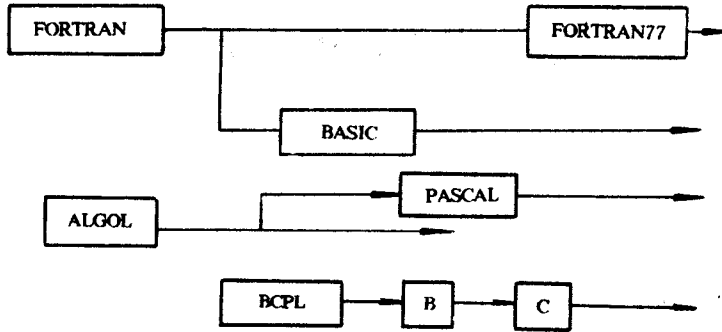


图 1-1 C 语言和其它语言的关系

(2) C 语言是一种结构化程序设计语言,即程序的逻辑结构可以用顺序、选择和循环三种基本结构组成。C 语言具有诸如 if~else、for、do~while、while、switch~case 等结构化语句,便于采用由顶向下、逐步细化的结构化程序设计技术。因此,用 C 语言编制的程序,具有容易理解、便于维护的优点。

(3) C 语言是便于模块化软件设计的程序语言。C 语言程序的函数结构,十分利于把整体程序分割成若干相对独立的功能模块,并且为程序模块间的相互调用以及数据传递提供了便捷的途径。这一特点也为把大型软件模块化,由多人同时进行集体性开发的软件工程技术方法提供了强有力的支持。

(4) C 语言具有种类丰富的运算符。除一般高级语言使用的 +、-、*、/ 这些四则运算及与 (AND)、或 (OR)、非 (NOT) 等逻辑运算功能外,还可以实现以二进制位 (Bit) 为单位的位与 (&)、位或 (|)、位非 (~)、位异或 (^) 以及移位操作 (>>、<<) 等位运算。并且具有如 a++、b-- 等单项运算和 +=、-=、*=、/= 等复合运算功能。

(5) C 语言程序中可以使用的如 #define、#include 等编译预处理语句,能够进行字符串或特定参数的宏定义,以及实现对外部文本文件的读取和合并。同时还具有 #if、#else 等条件编译预处理语句。这些功能的使用提高了软件开发的工作效率,并为程序的分块编译和程序的可移植性提供了强有力的支持。

(6) C 语言的数据类型丰富。基本类型有整型、实型、双精度实型、字符型、数组、指针等基本类型。同时,它还具有结构体和联合体两种结构类型,利用它们可以方便地处理包含着不同数据类型的复杂数据结构。因此,C 语言具有很强的数据处理能力。

(7) C 语言程序具有较高的移植性。C 语言的语句中,没有依存于硬件的输入与输出语句。程序的输入与输出功能是通过调用输入与输出函数实现的。而这些函数是由系统提供的独立于 C 语言的程序模块库,因此,C 语言程序本身并不依存于机器硬件系统,从而便于在硬件结构不同的机种间实现程序的移植。

三、C 语言的版本

◀必学/了解

最近几年中,适用于各种不同操作系统 (UNIX、MS-DOS 等) 和不同机种 (8 位~32 位) 的 C 语言编译系统相继出现,其种类达几十种之多。它们的语句功能基本上一致,可以圆满

地解决 C 语言程序在不同机种间的移植问题。但是, C 编译系统版本繁多, 也不可避免地造成了不同版本之间的某些差异, 它主要体现在标准函数库中收录的函数的种类、格式和功能上的差别。本书以当前最新的“1987 年美国国家标准 C 语言(87 ANSI C)”为基础, 同时兼顾目前广泛使用的 Turbo C, Borland C, Microsoft C 等版本的内容予以叙述。读者在使用 C 语言编制实用程序时, 最好首先参考所使用的计算机上配备的 C 编译系统的有关资料。

§ 1.2 C 语言总体印象

任何一种计算机程序语言, 都具有特定的语法规则和一定的表现形式, 程序的书写格式和程序的构成规则是程序语言表现形式的一个重要方面。我们必须强迫自己养成按照规定格式和构成规则书写程序的优良习惯, 这样可以使程序设计人员和使用程序的人易于理解。

一、C 语言程序的格式

◁必学/了解

在介绍 C 语言程序格式之前, 让我们首先回顾一下 BASIC 程序的格式, 然后通过两种语言程序的比较, 读者就会更直观地了解 C 语言程序的格式特点。

【引例 1】求圆柱体体积。

使用 BASIC 语言, 编制计算半径为 R, 高度为 H 的圆柱体体积的程序。要求 R 和 H 的数值由键盘输入。该程序如下:

```
10 INPUT R,H
20 V=3.14159*R*R*H
30 PRINT "V=",V
40 END
```

【引例 2】用 C 语言编写的具有同样功能的程序。

```
main()
{
    int r,h;
    float v;

    scanf("%d%d",&r,&h);
    v=3.1415926*r*r*h;
    printf("V=%f\n",v);
}
```

这里, 我们暂不讨论 C 语言程序中各个语句的功能, 首先把注意力放在其格式特点上, 比较上面两个程序, 可以看出, C 语言程序有以下若干格式特点:

(1) C 语言程序习惯上使用小写英文字母。这与 BASIC 程序习惯上使用大写字母正好相反。但是, C 语言程序中也可以用大写字母, 但它们常常是作为常量的宏定义和其它特殊用途使用, 关于这点将在本书后续章节中介绍。

(2) 和 BASIC 程序一样, C 语言程序也是由一个个的语句组成。每个语句都有规定的语法格式和特定的功能。在上面的 C 语言程序中, 通过与 BASIC 程序对照, 可以得知:

```
scanf("%d%d",&r,&h);
```

是用于输入变量数值的语句,此外,还有计算体积 v 的语句:

```
v=3.14159*r*r*h;
```

和输入输出语句:

```
printf("V=%f\n",v);
```

(3)和 BASIC 语言不同,C 语言程序不允许使用行序号。

(4)BASIC 程序中以回车符作为一行语句的终止符,以冒号(:)作为语句间的分隔符。与此相应,C 语言程序使用分号“;”作为语句的终止符或语句间的分隔符。对于初学者,这点常常被忽视,务必予以高度重视。

(5)一般情况下,每个 C 语言语句占用一个书写行的位置。更准确地说,C 语言程序不存在程序行的概念。一个程序可以自由地使用任意的书写行,即一行中可以有多个语句,一个语句也可以占用任意多行,但需要注意语句之间必须用“;”分隔。作为极端情况,上面的程序按下列格式书写,仍是正确的:

```
main(){int r,h;float v;scanf("%d%d",&r,&h);v=3.14159*r*r*h;printf("v=%f\n",v);}
```

把它输入计算机时,仍然能够正确接收和识别。

(6)C 语言程序中用一对大括号,表示程序的结构层次范围。一个完整的程序模块要用一对大括号表示该程序模块的范围,如【引例 2】程序中的第二行和最后一行的大括号对。此外,程序体中一部份,诸如 if、for、while、switch、else、do 等语句中,有时为了需要,这样的语句也要求用大括号对包围,以表示该语句的范围。

(7)C 语言中,为了增强可读性,可以使用适量的空格和空行。但是,变量名、函数名以及 C 语言本身使用的单词(在 C 语言中称为关键字,如 if、for、while、char、int 等),不能在它们中间插入空格。除此之外的空格和空行是可以任意设置的。

综上所述,C 语言程序的书写格式自由度较高、灵活性很强,有较大的任意性。但是,为了避免程序书写时层次混乱不清,便于人们阅读和理解,一般都采用有一定格式的习惯书写方法。这样的书写格式并不是计算机要求的,而是为了给人们提供便利。

本书中推荐使用的格式称为“紧缩对齐”格式,见【例 1.2-1】。

【例 1.2-1】统计输入文件中行、单词和字符数量的程序。

```
#include<stdio.h>
main()
{
    int c,nl,nw,nc,inword;

    inword=0;
    nl=nw=nc=0;
    while((c=getchar())!=EOF)
    {
        nc++;
        if(c=='\n')
            nl++;
    }
}
```

```

        if(c==' ' || c=='\t' || c=='\n')
            inword=0;
        else if(inword==0)
        {
            inword=1;
            nw++;
        }
    }

    printf("line=%d word=%d character=%d\n",nl,nw,nc);
}

```

由上述程序,可以归纳出“紧缩对齐”格式的要点是:

- (1) 一般情况下,每个语句占用一行。
- (2) 不同结构层次的语句,从不同的起始位置开始,在同一结构层次中的语句,都缩进同样的字数。如程序中的 while 和 if、else 语句,书写都缩进相同位置。计算机输入 C 语言源程序时,一般使用空格键或制表键来调整各行的起始位置。
- (3) 表示结构层次的大括号,写在该结构化语句第一或第二个字母的下方,表示该结构语句开始的“{”与表示该结构化语句结束的“}”在竖直方向上应对齐,并且分别占用一行(本书为了节省版面,在程序结构比较清晰的前提下,“{”与“}”有时并不占用一行,请读者注意)。如 while 下方的“{”和倒数第三行的“}”是表示 while 语句范围的大括号对。同样,else 下方的大括号对也是如此。
- (4) 语句中不同单词间可以加有空格,如 int 后面有一空格。程序不同部分的行间可以加有空行,如 int 下面可以有一空行。

采用“紧缩对齐”格式书写的程序,结构层次清晰,充分体现了结构化程序的特点,十分便于阅读和理解。

C 语言的几种格式与书写注意事项

◀ 自学答疑

本章中介绍了 C 语言程序书写时常用的“紧缩对齐”格式,然而,除了这种格式以外,为了节省版面,也采用其它的变通写法,请看下面的例子。

1. 标准的紧缩对齐格式

```

/* print string as uppercase */
#include <stdio.h>

main()
{
    char str[20];
    int i;

    gets(str);
    for (i=0;str[i]!='\0';i++)

```

```
    putupper(str[i]);
}
```

2. 变通格式之一

其特点是取消程序中不同语句部分的空行,另外“{”与“}”不再占用一行:

```
/* print string as uppercase */
#include <stdio.h>
main()
{ char str[20];
  int i;
  gets(str);
  for (i=0;str[i]!='\0';i++)
    putupper(str[i]);}
```

3. 其它变通格式

除了上述的格式外,在某些情况下还可以将较短的几个语句书写在同一行上,以求节省版面,例如:

```
main()
{
    int i,j,k;
    i=1;
    j=1;
    k=1;
    :
```

亦可写为:

```
main()
{
    int i,j,k;
    i=1;j=1;k=1;
    :
```

因为,前面已经说过,C语言不是以语句行来区分不同的语句,而是依靠语句间隔符“;”来区分不同的语句的。

另外,当一个程序语句在一行之内写不下时,可以拆分成多行写,例如:

```
if(a>=0&& b>=0&& c>=0)printf("%d,%d,%d",a,b,c);
```

可以写为:

```
if (a>=0&&b>=0&&c>=0)
    printf("%d,%d,%d",a,b,c);
```

【注意】拆分时,应注意拆分的位置,可以在一个关键字之后或一个完整的表达式之后进

行拆行,而不要在关键字之间,或一些易引起混淆而造成计算机识别错误的地方换行。例如,将上面提到的语句拆成:

```
if
(a>=0&& b>=0&& c>=0)
printf("%d,%d,%d",a,b,c);
```

是可行的,而将其拆成:

```
(1) if
(a>=0&& b>=0&
&c>=0) printf ("%d,%d,%d", a, b, c);
```

或

```
(2) if (a>=0&& b>=0&& c>=0) pri
ntf ("%d,%d,%d", a, b, c);
```

或

```
(3) if (a>=0&& b>=0&& c>=0)
printf ("%d,%d,%d
", a, b, c);
```

就不行,它们都无法被计算机正确识别。原因在于:

- (1) "&&"是一个运算符,不能将其拆开。
- (2) printf 是一个关键字,将其拆行将造成计算机识别错误。
- (3) "%d,%d,%d"是一个完整的说明符,不能分行书写。

总之,拆分应在不会使计算机误识别的一个完整的语法单位处进行,这一点,以后慢慢地会理解透彻的,现在不必强求。

此外,非要在一些不宜拆行处拆行时,可以采用续行符号"\来进行。

例如:

```
if (a>=0&& b>=0&& c>=0) printf ("%d,%d,%d", a, b, c);
```

可以拆分成:

```
if (a>=0&& b>=0 \
&& c>=0) printf ("%d,%d,%d", a, b, c);
```

必须指出,这种续行符号"\在 Turbo C 中才被允许使用。

还必须指出,现在的全屏幕编辑器能允许一行有 255 个字符,它比屏幕的宽度(80 个字符)大得多,因此,一条语句一般都可以在一行内写下,但在阅读时,必须卷动屏幕(一般用的是  或  键)来看清楚 80 个字符以后的内容。因此,限制一条语句宽度在 80 个字符以内将会有利于阅读程序。

二、C 语言程序的结构特点

◀必学/了解

一个完整的 C 语言程序是由一个或多个具有相对独立功能的程序模块结合而成,这样的程序模块称为函数。从格式上看,每个函数是由函数名和大括号对“{ }”包围的若干语句组成的。举例如下:

【例 1.2-2】体现 C 语言程序结构特点的示范程序。

```

/* print string as uppercase */
#include <stdio.h>
main ()
{
    /* 第 4 行 */

    char str [SIZE];
    int, i;

    gets (str);
    for (i=0; str [i] != '\0'; i++)
        putupper (str [i]);    /* 第 11 行 */
}                               /* 第 12 行 */

putupper (ch)
char cc;
{
    char cc;

    cc = (ch >= 'a' && ch <= 'z') ? ch + 'A' - 'a' : ch;
    putchar (cc);
}

```

对该程序说明如下：

(1) 可以看出，上述程序是由名字称为 main 和 putupper 的两个函数组成。在组成 C 语言程序的函数中，必须有一个且只能有一个名字为 main 的函数，它叫做主函数。除主函数之外的函数都可以由用户命名，如上例中的 putupper 函数。

(2) C 语言程序的执行是从主函数 main 开始的，主函数中的所有语句执行完毕，则程序执行结束。如上例的程序是从第 4 行“{”开始，执行到第 12 行的“}”结束。在执行第 11 行的语句时，程序控制转移到函数 putupper 中。执行完函数 putupper 的语句后，再返回主函数中继续运行。这个转移过程叫做调用函数 putupper。因此看出 C 语言程序中，main 函数之外的其它函数都是在执行 main 函数时，通过嵌套调用而得以执行的。在程序中除了可以调用用户自己编制的函数外，还可以调用由系统提供的标准函数，如【引例 2】中的 printf 和 scanf，以及【例 1.2-2】中的 gets 和 putchar 等都是标准函数。

(3) 程序前部带有 # 号的语句是编译预处理语句。

(4) 和其它程序语言一样，C 语言程序中也可以使用注解。注解部分的格式是：

```
/* 注解内容 */
```

如本例中的第 1 行、第 4 行、第 11 行、第 12 行就有注解部分。注解可以加在程序的任意部分，它可以占用一行以上的位置，也可以写在语句的后面。注解部分作为源程序的一个部分保留在源程序清单中，但对源程序进行编译时，系统将略过注解部分。适当地使用注解，对提高程序的可读性十分有益。在以后的程序中读者将会体会到这一点。

(5) C 语言函数定义的一般格式如下所示:

函数名 (参数)

参数说明;

{

 数据说明部分;

 执行语句部分;

}

有的函数不带参数, 则参数说明部分也就不存在了。但函数名后的圆括号不能省略。如【例 1.2-2】中 main 函数不带参数, 而 putupper 函数有一个参数 ch, 紧接函数名后的是该参数的说明。第 6、7 行和第 16 行分别是两个函数内的数据说明部分, 而第 9、10、11 行和第 18、19 行是执行语句部分。

关于 C 语言的注解

◁ 自学答疑

高级语言一般都有注解, 使用注解可以使程序意义明确, 使程序易懂, 易维护, 此外, 还可以得到有关程序的作者、编制完成的时间等信息。

【例 1.2-3】注解示例。

```
/* This function is getting */
/* the two number's summer */
int sum (x, y)
int x; /* x is a number */
int y; /* y is another number */
{
    return (x+y); /* get the total */
}
```

本例中的注解说明了函数的功能、参数的意义、函数运算的结果, 虽然我们还不能具体地知道该函数的每一条语句功能, 但是注解已经为我们指出了本函数的作用。

又如, 下面是一个输出 (打印) 汉字的程序, 我们目前无法仔细分析其中每条语句的作用, 但是, 下面的注解指出了本函数的大致功能, 还指出了本函数的作者、编制完成的时间等。

```
/* Author: XuYuan, 1995-3-1 */
/* Output a chinese character. This function uses chinese character
library. You can understand more about the chinese library through
reading the text. nm is a chinese character internal code.
The chinese character can show in 0 or 90 degree. */
void outcc16 (int x, int y, int nm, int color, int angle)
{
    unsigned int qu, wei; /* A chinese word qu and wei code */
    int i1, i2, i3, rec;
    long addr;
```