

计算机应用普及丛书

C语言速成

严洪华 沈建威 倪旭东 编著



YUYAN
SUCHENG

YUYAN
SUCHENG

YUYAN
SUCHENG



安徽科学技术出版社

C 语言 速 成

严洪华 沈建威 倪旭东 编著

安徽科学技术出版社

(皖)新登字 02 号

责任编辑: 胡正义

詹小杰

封面设计: 王国亮

内 容 简 介

本书是专门为自学 C 语言及其程序设计的读者编写的基础教材, 全书采用循序渐进的内容安排, 深入浅出的讲解方式, 并辅以各种类型的例题介绍, 从而使读者能通过对该书的学习, 迅速地掌握 C 语言的基本内容, 尽快地运用 C 语言来编写实用程序。

本书的作者将他们多年从事 C 语言程序设计的丰富经验贯穿于该书的编写之中, 使得该书结构合理、概念清晰、例题丰富实用、通俗易懂, 充分体现了 C 语言在程序设计中的高效、灵活、功能强大及入门容易等优点。本书在向读者介绍 C 语言基础知识及其程序设计方法的同时, 还向读者提供了一些有实用价值的 C 语言编程技巧及其源程序清单, 是学习 C 语言程序设计的理想教科书。

本书可供中、高等院校计算机和非计算机专业的学生、计算机工作者和爱好者、管理人员和其他工程技术人员阅读参考, 也可作为计算机培训班的教材。

严洪华 沈建成 倪旭东 编著

*

安徽科学技术出版社出版

(合肥市九州大厦八楼)

邮政编码: 230063

安徽省新华书店经销 巢湖地区印刷厂印刷

*

开本: 787×1092 1/32 印张: 13 字数: 33.2 万字

1995 年 7 月第 1 版 1995 年 7 月第 1 次印刷

印数: 6 000

ISBN 7-5337-1202-1/TP·18 定价: 11.50 元

本书配有 5.25" 高密软盘一张 定价: 15.00 元

(本书如有倒装、缺页等问题向承印厂调换)

前 言

众所周知,C语言是世界上应用最广泛的几种计算机语言之一。其主要特点是能以高级程序设计语言的结构和编程环境,给人们提供了类似于汇编语言那样的系统资源操纵能力及程序执行效率,从而使之成为开发计算机软件的极为有效之工具。

这就是说,C语言作为一种结构化程序设计语言,比其他高级语言更贴近计算机的硬件。同时,C语言又是一种编译性的语言,具有生成的目标代码紧凑且效率较高、程序运行速度较快的特点。它有着丰富的数据类型、运算功能,能直接进行位、字节、字和指针的操作。它所带的函数库中,包含输入输出、字符串处理、文件管理、数据转换、内存管理等一系列的应用函数。而且,它的指令语句较少,语法也不复杂,入门比较容易。所以深受广大计算机工作者的欢迎。

然而,C语言在具有高效、灵活、功能强大及入门容易等优点的同时,却给人们带来了“得道难”的问题。一般情况下,凡具有使用其他高级语言编程经验的计算机工作者,在一周时间里读完一本C语言应用手册并上机编制一些简单的程序,并非难事,但要真正掌握C语言的精髓,充分发挥它的潜力,做到应用自如却并不那么容易。即使是对C语言有一定了解的编程人员,在开发一个大、中型规模的实用系统软件的过程时,也常常会被一些突如其来且难以解释的错误搞得头昏脑胀。

当然,“入门”既不难,只要方法得当,且持之以恒,“得道”也是可以做到的。我们编写本书之目的就是希望在向读者介绍C语言基础知识及其编程方法的同时,能为大家提供一些有实用价值的编程技巧,以便对读者在解决“得道难”的问题上能有所裨益。

我们为达到上述目的所采用的方法是:在介绍C语言基础知识的同时,精心安排一些简便实用的例程;在介绍C语言程序设计时,则尽量多补充介绍一些既有一定特色又有实用价值的源程序清单。例如在本书最后一章里,我们就向读者提供了一个用C语言自行编制的“一次读盘多次写盘的软磁盘全盘拷贝软件”的源程序清单。它的大部分操作指令都有中文注释。不言而喻,研究与运行别人编制的源程序是深入学习C语言的一个好方法。因为一个精心编制的程序不仅是一个算法、数据结构、程序设计方法和语言运用的综合体,同时也是一种良好的编程技术与风格的体现。

本书共分为12章,第一章对C语言的由来、特点及基本词法等作一简单介绍。第二章则讲述C语言中的常量、变量类型、运算符与表达式的应用。第三章介绍了C语言的语句与控制结构。第四章则介绍了C语言的重要数据类型——数组。在第五章里我们用了较多的篇幅来重点讲解了一下C语言中另一个重要的数据类型——指针。第六章介绍了C语言数据类型——结构。第七章介绍了一下联合、枚举及变量的自定义类型语句 typedef 的应用。第八章详细了讲解C语言的重要成分——函数。第九章讲解了C语言的计算机设备终端的输入输出与磁盘文件的处理。第十章简要地介绍一下C语言的屏幕与图形处理功能。第十一章则简要地介绍一下C语言的预处理及编译与连接操作。第十二章简要介绍C语言的程序设计,并以我

DJS 11/10/05

们自行编制的“一次读盘多次写盘的软磁盘全盘拷贝软件”(MYCOPY)为例进行讲解。在本书的最后一部分,以附录方式向读者提供MYCOPY的全部源程序清单。该清单的大部分语句后面都加了中文注释,以帮助读者比较容易地了解各指令的含义。只要将这个程序的内容在IBM PC及其兼容机上正确输入,并通过Turbo C 2.0以上版本的系统编译、连接成可执行文件,得到的就是一个非常实用的一读多写磁盘拷贝软件。您可以在DOS操作系统下用该软件进行磁盘的批量复制,即读一次原盘可以复制出任意张相同的目标盘。它还可以对720K、1.2兆、1.44兆等大于计算机基本内存容量(一般为640K)的磁盘,在计算机配有硬盘的情况下,一次读入原盘,并一次复制完目标盘的操作,不受计算机内存的限制,中途也不必更换原盘和目标盘。同时,该程序还提供一个可选择驱动器和磁盘类型的下拉式菜单,以便用户选择不同驱动器进行各类常用磁盘的复制,从而使您能顺利地完成磁盘的拷贝工作。

为了方便读者,我们还将MYCOPY的源程序清单及其编译后的执行文件全部录入到一张软磁盘中,使之与本书配套提供给大家,需要者可直接与安徽科技出版社联系。

需要说明的是:本书介绍的C语言是其最基本的部分,所以C语言中一些比较高深或比较冷僻的内容难以在这本教材中全部涉及,但是作为C语言的核心和主要内容,即C语言常规编程中必不可少的东西,本书还是作了比较详尽的介绍。由于计算机技术的飞速发展,软件设计语言和方法一直在不断完善与更新,加上我们水平的限制,编写的内容难免会有疏漏与失误之处,恳请读者指正。

作 者

一九九五年四月

目 录

第一章 C语言综述	1
第一节 C语言的由来及特点	1
一、C语言的由来	1
二、C语言的特点	2
第二节 一个简单的C语言例程的编译与运行	2
第三节 C语言的基本词法	4
一、C语言的基本词汇	4
二、运算符	5
三、转义字符	5
四、标识符	5
五、关键字	6
第四节 C语言程序的基本结构	6
 第二章 常量、变量、运算符与表达式	10
第一节 C语言的基本数据类型	10
第二节 常量	11
一、直接常量	11
二、符号常量	12
第三节 变量	12
一、变量的定义	12
二、变量的存储类型	13
三、变量的初始化	16
第四节 运算符和表达式	17
一、算术运算符和表达式	17
二、按位运算符和表达式	18
三、赋值运算符和表达式	18
四、数据类型强制转换符和表达式	19
五、关系运算符和表达式	19
六、逻辑运算符和组合关系表达式	20
七、求类型长度运算符 (sizeof)	20
八、逗号运算符和表达式	20
九、条件运算符和表达式	21

第三章 程序控制语句	23
第一节 条件分支语句	23
一、if~else 语句	23
二、if 语句	24
三、条件分支语句的嵌套	24
四、多重条件语句	26
第二节 循环控制语句	27
一、for 语句	27
二、while 语句	30
三、do~while 语句	31
第三节 开关语句与跳转语句	32
一、switch (开关语句)	32
二、goto (无条件转移语句)	34
三、break (中断语句)	36
四、continue (条件继续语句)	36
五、return (返回语句)	37
第四节 分程序结构语句	37
第四章 数组	40
第一节 一维数组	40
第二节 字符数组	42
第三节 多维数组	44
第四节 数组的初始化	46
第五章 指针	49
第一节 指针与地址	49
第二节 指针与数组的关系	51
第三节 指针的初始化	53
第四节 指针变量的基本运算	55
一、取地址运算和取内容运算	55
二、指针变量值加或减一个整数	55
三、两个指针变量相减	56
四、用关系运算符对两个指针变量进行比较	56
第五节 内存动态分配	56
第六节 指针数组和多级指针	58
第六章 结构	61
第一节 结构的定义	61
第二节 结构成员的引用	64

第三节 结构数组	66
第四节 指向结构的指针	68
第五节 链表	70
第七章 联合、枚举和自定义数据类型	78
第一节 联合	78
第二节 枚举	81
第三节 自定义数据类型	84
第八章 函数	86
第一节 函数的定义与调用	86
一、函数的定义	86
二、函数的调用	87
第二节 返回语句	89
一、从函数返回	90
二、返回值	91
第三节 函数的作用域规则	91
第四节 函数的参数	91
第五节 传值调用与传址调用	92
一、传值调用	93
二、传址调用	94
三、用数组作函数的传递参数	95
四、用指针作函数的传递参数	97
五、用结构作函数的传递参数	98
六、命令行参数	100
第六节 递归调用	103
第七节 函数指针	106
第八节 库函数	108
第九章 输入、输出及磁盘文件	110
第一节 标准设备文件的输入输出	110
一、getchar 和 putchar	111
二、gets 和 puts	111
三、格式化的终端输入输出函数 printf 和 scanf	113
第二节 一般文件的输入输出	118
一、fopen	119
二、fclose	121
三、putc	121
四、getc	121
五、fgetc	122

六、fputc	123
七、fprintf	124
八、fscanf	125
九、getw	125
十、putw	126
十一、feof	126
十二、rewind	127
十三、fseek 和随机访问输入输出	127
十四、fread	128
十五、fwrite	128
第十章 图形	131
第一节 图形绘制基本函数的编程	131
一、显示方式与调色板的设置	131
二、画点	133
三、画线	134
四、画矩形	134
五、画圆	134
六、区域填充	135
第二节 一个绘图程序实例	135
第十一章 C 语言程序的预处理与编译	139
第一节 C 语言的预处理	139
一、#include	139
二、#define	140
三、#undef	142
第二节 C 语言程序的编译	142
一、使用 Microsoft C 编译系统的操作	143
二、使用 Turbo C 编译系统的操作	144
第十二章 C 语言程序设计及应用举例	147
第一节 C 语言程序的开发过程	147
第二节 C 语言程序设计的应用举例	149
一、MYCOPY 下拉式系统主菜单显示与操作的编程方法	149
二、磁盘拷贝模块的编程	161
附录一 MYCOPY 源程序清单	173
附录二 C 语言编译系统的常用库函数	193

第一章 C 语言综述

C 语言作为一种通用的程序设计语言,由于它的紧凑、高效、易于移植、模块化等特点已被广泛地应用于系统程序设计、控制工程、科学计算、数据分析、图形处理、文字编辑以及人工智能等领域,而且越来越为广大的程序员所喜爱。本章主要介绍 C 语言的一些最基本的知识,尽管这种简略介绍很难体现 C 语言的上述优越性,但它却可以使读者对 C 语言有一个初步的认识。由于这一章所讲内容的大部分,我们在以后的章节里将会重新进行详细介绍,所以读者对本章内容的一些细节可以先不必深究。

第一节 C 语言的由来及特点

一、C 语言的由来

C 语言是 UNIX 操作系统的主程序设计语言。我们知道,最初的 UNIX 系统是用汇编语言编写的。但由于汇编语言的移植性、程序开发效率及易读性都不如高级语言,因此人们希望有一种既能象汇编语言一样能对硬件进行操作,又具有高级语言可读性、移植性强特点的新语言。C 语言就是适应这种需要而产生的。现在的 UNIX 操作系统百分之九十的程序和绝大多数实用例程以及工具,都是用 C 语言开发的。

1970 年美国贝尔实验室的 K. Thompson 对一种比较贴近硬件的语言 BCPL 进行了简化,设计出 B 语言,并用它改写了 UNIX 操作系统。后来为了克服 B 语言过于简单、功能不强和生成的解释代码运行速度慢的弱点,1972 年贝尔实验室的 D. M. Ritchie 在 B 语言基础上设计了 C 语言,第二年他和 K. Thompson 合作又用 C 语言重写了 UNIX 操作系统。用 C 语言改写的 UNIX 操作系统代码量比以前的版本扩大了三分之一,加进了多种程序设计功能。现在使用的 UNIX 版本都是在此基础上扩充和发展起来的。

七十年代末,随着微型机的迅速发展,C 语言开始移植到非 UNIX 系统上。各种应用于微机的非 UNIX 的 C 编译语言版本陆续出现,C 语言脱离了 UNIX 成为一个独立的通用程序设计语言。

多年来,人们在实现各自的 C 编译时,是参照 1978 年 Kernighan、Ritchie 的《C 语言参考手册》中的定义。本书后面一些地方提到的“老版本”,就是指以这本书为基础的 C 语言版本。但在 C 语言的发展过程中,由于各自实现环境的不同,导致了 C 编译版本的庞杂,不利于 C 程序之间的交流。1983 年,美国国家标准协会 ANSI 在继承和发展的基础上开始着手制定新的 C 语言标准。本书着重介绍的 Microsoft C 和 Turbo C 已实现了大部分 ANSI 标准的规定,并作了部分扩充,这是当前微机上应用较为广泛的两种 C 语言版本。不言而喻,C 语言是在实践中产生和发展起来的,它必然还将在实践中继续发展和完善。

二、C 语言的特点

相对于其它计算机语言，C 语言有着下列明显的优点：

1、C 语言是一种表达能力强，程序设计的自由度大，特别适用于程序员使用的计算机语言。它能对计算机硬件进行操作，能直接访问物理地址和进行字位操作，它对数组越界和函数调用时参数匹配一般不作检查，而且在低层的基本数据类型可以任意转换，它具有高级语言的易读、易写、易查错、易维护的特点。也就是说，C 语言能以高级程序设计语言的结构和编程环境，给人们提供类似于汇编语言那样的系统资源操纵能力及程序的执行效率，从而使之成为开发计算机软件的极有效的工具。是深受计算机软件设计人员欢迎的一种程序设计语言。

2、C 语言代码紧凑、高效，书写简单、易学，其程序的目标码执行效率仅比汇编程序低 10~20%。C 语言仅有四种基本数据类型，32 个关键字，语法规则也不复杂。小巧的核心不仅使得学习 C 语言相对简单，而且使得其编译程序也比较小，因而可以方便地在微型计算机上开发系统和应用软件。

3、C 语言提供了多种结构化的控制语句，是理想的结构化语言。C 语言程序文件可以分割成几个源文件分别进行编译。它以分立的函数为基础能够实现程序的模块化，加上它有大量的库函数支持和可以由基本数据类型构造出各种复杂的组合数据类型，所以特别适合于大型程序的模块化设计，因此 C 语言是一种具有现代风格的程序设计语言。

4、C 语言有很好的移植性。大部分代码不改动就可把编译程序从一个机器上移植到另一个机器上。C 语言的众多微机版本快速形成产品也归功于它的可移植性。随着近年来 C 语言标准化的开展，C 语言的可移植性更有保障。

5、C 语言有丰富的运算符和设计结构，灵活地运用它们可以实现各种复杂的运算，它对各种不同类型程序设计有很好的适应性。

当然 C 语言和其它计算机语言一样也存在着一些不足。C 语言最大的不足在于其安全性，这是由于 C 语言的数据和指针没有屏障，类型转换比较随便，对数组越界及变量类型一致性不作语法上的严格检查等原因所造成的。然而安全性虽是最大但又不是非克服不可的缺点，只能以辩证的观点来看待它。1988 年 ANSI 的 C 标准，对 C 的安全性作了增补，但始终不把 C 改造为强类型的安全语言，旨在保持 C 语言高效、灵活和编程快速的特色。实际上安全性很大程度上取决于程序员的编程风格和水平，因此要强调由程序员自己去保证程序的正确性。

第二节 一个简单的 C 语言例程的编译与运行

C 语言是一种以编译方式运行为主的程序语言。它也有解释方式下运行的系统，即也可在解释方式下运行，但通常仅作为调试工具用。所以一般的 C 语言的源程序，必须先经过编译程序把源程序变成机器可直接执行的代码，然后执行。由 C 源程序的编写直到可执行代码的产生，中间需要以下几个步骤：

- 1、由编辑程序产生源程序文件；

- 2、对源程序文件进行编译，形成目标文件；
- 3、把目标程序同必要的库函数连接在一起，形成可执行文件；
- 4、执行程序。

在目前使用的 Microsoft C 和 Turbo C 等 C 语言编译系统中已将第二、第三步合并用命令行编译方式就完成了。

下面通过一个最简单的 C 语言程序来说明一下它是如何编译和运行的，以便让读者能对 C 语言程序有一个初步的概念：

例1.1 在屏幕上打印字符串“欢迎你参加 C 语言的学习!”。

先用字处理软件编辑一正文文件 HY.C；

```
main ()                                /* 主函数，是执行时的起始点 */
{
    printf (" 欢迎你参加 C 语言的学习!");
    /* 用 C 语言系统提供的输出库函数 printf 把双引号包括的内容输出到屏幕上 */
}
```

再用 Microsoft C 编译系统在命令行方式下用 CL.EXE 文件对 HY.C 源程序进行编译与连接的操作，使之产生目标文件 HY.OBJ 并生成可执行文件 HY.EXE。

```
C>CL HY.C <CR>
```

最后在 DOS 状态下运行 HY.EXE 文件，结果为：

```
C: >HY <CR>
欢迎你参加 C 语言的学习!
```

其中<CR>为输入回车键。HY.C 是源程序文件名。C 语言源程序文件名必须带有“.C”的扩展名。关于 C 语言程序的编译和连接的操作方法，本书的第十一章将要详细介绍，这里从略。

该例只有一个功能，即在屏幕上打印一个字符串。从例程中可以看到，C 语言是由若干个称为“函数”的段组成的，函数是 C 语言的基本程序单位。本例中函数 main 是程序的主函数。每个 C 语言程序都必须有一个主函数，C 语言规定这个主函数必须命名为 main。主函数 main 的第一条可执行语句即为此程序的开始点，main 通常采用直接调用其它函数的方式控制着整个程序的执行。程序的停止点通常是 main 函数的最后一条可执行语句，当然有时也可能停止在程序中的其它语句上，这取决于执行的具体情况。程序里的一对花括号“{”和“}”标志着函数体的开始和结束。

在一个 C 语言程序里至少包含一个函数，即 main 函数，但在通常情况下，C 语言的源程

序总是还包含多个其它函数，它们可以是系统提供的标准库函数，也可以是由编程者自己编制并命名的函数。这些函数可以看成相当于其它语言中的子程序。一个函数名后必须跟一对圆括号“(”和“)”，括号里可以带若干个参数，本例中的 main 函数没有带参数。在例程中还有一个函数 printf，这是个由系统提供的库函数，它的作用是向系统指定的输出设备按指定格式输出若干个数据信息。在本例中，printf 把双引号间的一个字符串参数输出到计算机屏幕上。

在上述例程中，我们还可以看到 C 语言的语句都是以分号“;”作为结束符的。C 语言是一种不强调书写格式的程序语言，它的书写是比较自由的。一个语句可以分几行写，也可以把用分号分隔的几个语句写在同一行。程序中“/*”和“*/”之间的部分为注释，它是供程序员看的，对程序的编译和运行不起作用。注释的个数不受限制，它可以放在源程序文件里任何允许插入空格的地方。ANSI 标准不支持嵌套注释，因此，在一般情况下不要使用嵌套注释。

第三节 C 语言的基本词法

C 语言的基本词汇包括字母、数字以及对 C 语言编译系统有特殊意义的标点符号。C 语言以这些基本词汇构成基本语法单位，包括标识符、关键字、分隔符、运算符和常量，C 语言程序就是通过这些字符组合成为有意义的语句加以构造的。

一、C 语言的基本词汇

1、字母、数字和下划线

大写英文字母 A~Z、小写字母 a~z、阿拉伯数字 0~9 及下划线可以用来组成后面将要介绍的常量、标识符和关键字。

需要注意：C 语言编译系统把大写字母和小写字母当作不同的字符。如果在一个给定的项中规定一个小写字母 a，在使用这个项时就必须使用该小写字母，而不能用一个大写字母 A 代替。

下划线起一个字母的作用。所以在构成标识符等语法成份时，可以以下划线打头，这在其它很多高级语言中是不允许的，而在 C 语言中则是合法的。

2、空白符

空格符、水平和垂直制表符、回车符、换行符、换页符都称为空白符。在程序中这些字符将用户定义的项，例如常量和标识符，同其它的项分隔开来。另外，在 C 语言中注释也被当作空白符。C 语言编译系统忽略各空白符，除非它们用作分隔符或者作为字符串的成分。这意味着可以使用额外的空白符来增加程序的可读性。

3、标点符号和特殊符号

C 字符集中的标点符号和特殊字符有多种用途，有些用于从组织程序文本到定义编译系统或编译的程序所要执行的任务，如代表或组合成各种运算符；有些则只能出现在字符串常

量和注释中。在本书后面的章节里将会陆续讲到它们。

二、运算符

运算符是一些特殊字符的组合，它们规定对数值如何进行比较、转换和赋值。另外还有一个特殊的运算符 `sizeof`，用于测量一个数据类型的长度字节数。运算符有规定的优先级和结合特性，这些将在下一章再作详细的介绍。

三、转义字符

转义字符是个特殊的字符组合，它们在字符串和字符常量中表示空白符和非字符。它们的典型用法是规定终端和打印机的动作，如回车和制表动作，以及提供通常有特殊意义的字符的文字量表示，如双引号字符。转义字符由一个反斜杠及跟着的一个字母或者若干个数字组成。下表列出了一些常用的 C 语言转义字符。

表1.1 转义字符序列表

转义字符	含 义	转义字符	含 义
<code>\t</code>	水平制表	<code>\'</code>	单引号
<code>\v</code>	垂直制表	<code>\"</code>	双引号
<code>\b</code>	退格	<code>\\</code>	反斜杠
<code>\r</code>	回车	<code>\ddd</code>	八进制数表示的 ASCII 字符
<code>\f</code>	换页	<code>\xhh</code>	十六进制数表示的 ASCII 字符

如果在一个反斜杠后面有一个不在上述表中的字符，则忽略反斜杠而将该字符表示为文字常量。例如：在字符串常量或字符常量中，`'\c'`表示字符 `c`。

四、标识符

标识符是程序中所使用的常量、变量、函数和数据类型等标号的名字。例如可以生成一个标识符，用它说明一个变量或函数，然后在程序的其它语句里使用该标识符来引用该项变量或者函数。

标识符是一个字符序列，它必须以一个英文字母或下划线开头，后面跟零个或多个字母、数字或下划线。一个标识符允许有任意多个字符，但在不同的系统上，标识符的有效字符数是不同的。ANSI 的 C 标准规定标识符的有效位数可达31个字符。

选取标识符应尽量做到简洁、易识别，并能尽可能表达其显见的含义，以增强程序的可读性。打头的下划线必须小心使用，因为习惯约定以下划线打头的标识符通常是作为系统例程专用的。下列为合法的标识符的例子：

a p3 write_char MAX prog_12 _sum

但下面的标识符则是不合法的：

5a &H write--char pe. a

由于 C 语言编译系统将大写字母和小写字母当作不同的字符。因此，大小写字符不同的组合可以产生不同的标识符。例如，标识符 CH、Ch、cH 彼此就是不同的。

五、关键字

关键字是系统为特定目的而保留的一些单词，它们对 C 语言编译系统有着特殊含义。C 语言将这些关键字定义成数据类型符、语句定义符和存储类型符。它们只能按定义加以使用，而不能当作常规的标识符使用。所以，在程序中选用变量名、函数名等标识符时，不能同关键字发生冲突。ANSI 的 C 标准规定了以下由小写字母组成的关键字：

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	while
do	if	static	volatile

除此之外，一些 C 语言版本如 Turbo C 等还增添了一些别的关键字。关键字不能重新定义，但是通过使用某些指令，可以规定用其它文本符号来取代一些关键字。

下面七个字虽然不属于关键字，但是 C 语言预处理程序中的专用词。所以建议读者把它们看成关键字，不要在程序中随意使用，以免造成混淆：

define undef include ifdef endif line

第四节 C 语言程序的基本结构

在第二节里，我们用一个十分简单的 C 语言程序介绍了一些 C 程序的基本要素。下面再举一个例子来分析一下 C 语言程序的基本结构：

例1.2 比较输入的两个整数大小，在屏幕上显示大的那个数。

```

#include <stdio. h>                /* 预处理命令，输入输出函数说明 */
int z;                             /* 定义外部变量 */
int max ();                         /* 外部函数说明 */

main ()                            /* 主函数 */
{
    int x, y;                       /* 定义局部变量 */
    scanf ("%d : %d", &x, &y);    /* 调用 scanf 库函数接收键盘输入 */
    z=max (x, y);                  /* 调用函数 max，返回值赋给 z */
    printf (" 大数是 : %d", z);    /* 在屏幕上打印变量 z 的值 */
}

int max (a, b)                     /* 函数定义 */
int a, b;                          /* 函数形式参数说明 */
{
    if (a>b) return (a);           /* 如果 a>b，返回 a 的值 */
    else return (b);              /* 否则，返回 b 的值 */
}

```

在例程中，最前面一句是预处理程序（本书的第十一章将具体介绍其用法）。第二、三句分别为外部变量说明语句和外部函数说明语句。和例1.1中一样，main 是主函数，在其函数体内定义了两个局部变量 x 和 y，并调用库函数 scanf 接收键盘输入的两个整数来给它们赋值。main 函数中有两句执行语句：一句是调用函数 max，并将函数的返回值赋给变量 z，括号中的 x、y 是传递给函数的参数，称为实参。另一句是调用库函数 printf 在屏幕上根据格式转换说明代码 %d 的要求用十进制方式打印变量 z 的值。程序下面部分定义了函数 max，括号里的 a、b 是和调用该函数时的实参相对应的参数，称为形参。控制 max 函数执行流程的是一组复合条件语句：如果形参 a 大于 b，函数 max 返回 a 的值，否则返回 b 的值。控制语句 return 是一个返回语句，它返回一个值给调用函数，并把程序的控制权交还给调用程序。

上述程序大体上反映了一个 C 语言源程序的主要结构。图1.1是 C 语言程序的基本结构示意图：

预处理语句
全局数据描述

```

main ()
{
    局部数据描述
    执行语句
}

```

```

函数1 ()
{
    局部数据描述
    执行语句
}

函数2 ()
{
    局部数据描述
    执行语句
}

.....

函数 N ()
{
    局部数据描述
    执行语句
}

```

图1.1 C 语言程序基本结构示意图

通过以上程序的介绍可以看出，C 语言程序是由若干函数组成的系统。因而学习 C 语言，主要要掌握两个方面：一是编写正确有效的新函数；二是正确有效地使用 C 语言系统的标准库函数。

在 C 语言的程序设计中，要习惯于用多个小函数来构成一个大程序，即尽量把一个大程序分解成若干个各自完成一个独立功能的小模块。我们可以把这些小功能模块看成“零件”，需要时就可以用这些“零件”组成“机器”，而不必每次都重新做起。

各函数在程序中排列的位置并不十分重要，main 函数也可以放在其它函数的后面，但程序总是从 main 函数的第一条语句开始执行。关于预处理命令、变量、函数和参数等将在以后的章节里陆续向大家介绍。为了方便下几个章节的讲解，我们有必要将该例程中用的两个库函数 printf 和 scanf 在这里先作一个简略介绍。

库函数 printf 和 scanf 又称格式化的终端输入输出函数，它们可以用你控制的各种格式读、写数据。其中格式化输出函数 printf 的一般形式为：

```
printf (" 格式命令", 变量元素表);
```

格式命令可分为两类，第一类是需原样输出的字符串内容；第二类包括定义输出方式的格式命令。这些格式命令由百分号开头，后面跟着格式码。变元表中的变元数与格式命令数完全相等，格式命令与变元按顺序匹配。如例1.2中的 printf (" 大数是： %d", z); 语句将在屏