

微型计算机基础知识教育丛书

吕凤嘉 编著

C 语言入门



2
/3

北京大学出版社

PUP

微型计算机基础知识教育丛书

C 语 言 入 门

吕凤翥 编著

北京大学出版社
北 京

新登字(京)159号

图书在版编目(CIP)数据

C语言入门/吕凤翥编著. —北京:北京大学出版社,
1994.8

(微型计算机基础知识教育丛书)

ISBN 7-301-02494-0

I. C… II. 吕… III. C语言—基本知识 IV. TP312C

书 名: C语言入门

著作责任者: 吕凤翥

责任编辑: 杨锡林

标准书号: ISBN 7-301-02494-0/TP·212

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

电 话: 出版部 2502015 发行部 2559712 编辑部 2502032

排 印 者: 蓝地公司激光照排 飞达印刷厂印刷

发 行 者: 北京大学出版社

经 销 者: 新华书店

版本记录: 787×1092毫米 32开本 11.125印张 246千字

1994年8月第一版 1994年8月第一次印刷

定 价: 9.50元

内 容 简 介

本书系统扼要地讲述了C语言的词法和语法规则。重点讲述C语言特点和其他高级语言的区别。内容由浅入深,语言通俗易懂,例题针对性强。每章后边都附有思考题、练习题和上机操作题,有利于帮助读者掌握和理解C语言的基础知识,适于自学。本书共有8章,主要内容包括:常量和变量、运算符和表达式,语句、函数、指针、结构以及文件。通过学习本书可以较快的理解C语言的基本语法知识并掌握一定的编程技巧,为以后继续学习打下基础。

本书适合作为各类大专、中专理工类学生的教材或参考书,也可供广大微机爱好者学习C语言的参考书。

序 言

从第一台电子计算机问世到今天，几近半个世纪，人类从生产到生活发生了巨大的变化，电脑已悄然闯入社会生活的各个领域。过去说：没有电将寸步难行；现在要说：没有计算机就没有现代化。

计算机科学是信息科学的一个重要组成部分。21世纪将以信息技术为主导，使整个社会的经济活动方式与社会的就业结构产生非常大的变化。体力劳动的比重将逐渐减少，掌握信息技术的脑力劳动者的比例将不断增大。电子邮件、电子新闻、电子图书等新的科技将逐步取代纸笔和印刷机，新的计算机文化将迅速发展。

著名的计算机科学家 G·伏赛斯曾预言：电脑将是继自然语言、数学之后而成为第三位的，对人的一生都有大用处的“通用智力工具”。现在，实践已经证明了电脑已经成为各行各业的基本工具。许多部门已经把具备电脑的应用知识与技能作为录用或考核工作人员的一个重要条件。综合国力的竞争说到底还是掌握高科技人才的竞争。怎样将计算机科学知识迅速而有效地普及到全社会，也就成了一件具有紧迫感的新任务。

近年来为适应社会的需求，各类职业教育学校有了较快的发展。在这些学校里的学生理所当然地要接受计算机教育。但是，目前的状况是，适用于这些学校的教材却非常之少。因此，尽快写出这种教材供同学们选用，是我们编写“微型计算机基础知识教育丛书”的初衷。从教学目标出发，这套丛

书将重点讲述基本概念和基本方法，以理论联系实际思路介绍一些具体的实际操作技术；在写作手法上，力求通俗而不肤浅，深入而不玄奥，贯彻循序渐进的原则；在每一应知应会的知识点上，着力讲深讲透；书中附有必要的思考题和上机练习题，引导读者既动脑又动手，学深，学活，学以致用。

随着电脑应用的普及，蒙在电脑上的一块神秘的面纱已经被揭落。许多学过电脑的人都感到，入门不难，深造也是办得到的。只要功夫深，电脑不会不听命。

**中国计算机学会普及委员会主任
清华大学计算机科学与技术系教授**

吴文虎

1994. 3. 25

前 言

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高级语言，它的历史仅有 20 余年，但它的声誉却不亚于它的先辈们。因此，我们说 C 语言不愧为高级语言中的一个新秀。

C 语言和其他高级语言一样，都遵循着计算机高级语言的一些共同的规律，但它继承了其他语言的优点，克服了一些语言的短处，并形成了它独有的风格。C 语言一问世就与 UNIX 操作系统结下了不解之缘。C 语言的简练、灵活、功能强、效率高受到了越来越多的人的青睐，特别是在系统开发方面、事务管理方面和图形图像等方面，深受人们的欢迎。因此，许多用过 C 语言的人们都认为 C 语言是当今高级语言中的佼佼者。

C 语言是一种面向过程的结构化程序设计语言。结构化程序设计是计算机软件发展中的一个里程碑，它比以前的非结构化程序设计是一大进步。人们今天用结构化程序设计方法进行程序设计要比以前用非结构化程序设计方法不知方便多少倍。这也是 C 语言具有强大的生命力的原因之一。而今，90 年代是面向对象程序设计的年代，一种面向对象的程序设计方法正在风靡全球，作为这种程序设计方法的主要语言是 C++，而 C++ 是 C 语言的全集，学会了 C 语言，掌握 C++ 便容易得多。

今天，越来越多人在学习 C 语言、使用 C 语言，用 C 语

言开发各个领域的应用软件，C语言已经是当今在世界上最流行的语言之一。因此，我认为学习使用计算机，除了学习简单的使用方法，学会进行汉字处理，接着就该学会使用C语言编程。不可否认BASIC语言是一种历史悠久、发展成熟、学习容易、使用简单的一种基础语言，很多人都是从学习BASIC开始学习使用计算机的。也不可否认BASIC语言至今还在不断发展，不断自我完善，不断推出新的版本和品种，也在向结构化方面发展。但是，与C语言比较它毕竟有些先天不足，只能后天弥补。

BASIC语言比较简单，容易学习，初学者比较喜欢。C语言比较复杂，学起来有困难，这是人们的共识。但是，C语言也不是那么神秘，也并不那么难学。它也是入门不难，深造也是办得到的，但是真正掌握C语言，能够运用自如，这需要一段实践的时间。BASIC语言可以作为初学者的一种语言，C语言也可以作为初学者的语言。一些有胆量的初学者不妨学一下C语言试一试，看一下自己是否能够入门。

《C语言入门》一书就是为那些初学计算机语言的读者编写的。作者相信不会太久，学习计算机操作系统从Windows开始，学习计算机语言从C开始。由于这本书的对象是初学计算机的读者，因此，书中尽量少用较深的计算机的词语，遵循一种高级语言的共同特点，强调写出C语言的风格。从浅入深，由简到繁。为使读者对C语言有个比较全面的系统的了解和认识，照顾到C语言的完整性；为了使读者学过后能够用于实际的工作中，注重实践，做到理论联系实际，既使读者获得“鱼”，又使读者获得“渔”。有了“鱼”，可以美餐一顿；有了“渔”，可以美餐一辈子。

该书的全部例题都是在16位微机用Turbo C 2.0版的

编译系统编译通过的。该书中尽量选用一些对训练讲过内容有帮助的例题，例题的覆盖面较大，难度较低。读者可以从例题中学到一些编程方法，积累一些编程经验。所用例题在计算方法上没有太大难度，中学程度足矣。

本书共分八章。第一章通过一个 C 语言编程的具体实例，了解 C 语言程序结构。引出本书下面章节的安排。第二章讲词法；第三章讲语句；第四章讲函数。这三章由单词、语句到函数是 C 语言程序的基本构成。第五、六两章讲 C 语言的精华部分：指针、结构，这是 C 语言中的难点，也是重点。第七章讲文件，作为 C 语言编程内容的结束。最后一章讲 C 语言的特点和应用，作为对 C 语言的一个小结，同时又向读者指明深造的方向。

本书每章后边都备有思考题、练习题和上机操作题，帮助读者更好地掌握本章内容，检查学习情况，更好地复习一些重要的概念和方法。

本书是作者在以前几本 C 语言著作的基础上改写的，力求通俗易懂，适于自学。由于本人水平有限，难免有错，多请指正。

作 者

1994. 6. 5

目 录

第一章 实例	(1)
1.1 一个例子	(1)
1.2 程序分析	(2)
1.3 程序的实现	(13)
1.4 程序的书写格式	(14)
思考题	(15)
练习题	(16)
上机练习	(17)
第二章 词法	(19)
2.1 字符和字符集	(19)
2.2 单词	(20)
2.3 常量和变量	(25)
2.4 运算符和表达式	(39)
2.5 类型转换	(67)
2.6 作用域和存储类	(70)
本章小结	(80)
思考题	(81)
练习题	(83)
上机练习	(90)
第三章 语句	(93)
3.1 预处理语句	(93)
3.2 说明语句	(107)
3.3 执行语句	(112)
本章小结	(137)

思考题	(144)
练习题	(144)
上机练习	(152)
第四章 函数	(154)
4.1 函数的定义和说明	(154)
4.2 函数的类型和返回值	(157)
4.3 函数的参数和函数的调用	(159)
4.4 函数调用的实例	(162)
4.5 系统提供的函数	(175)
本章小结	(179)
思考题	(182)
练习题	(182)
上机练习	(190)
第五章 指针	(191)
5.1 指针的概念	(191)
5.2 指针的运算	(195)
5.3 指针和数组	(198)
5.4 指针和函数	(213)
本章小结	(216)
思考题	(218)
练习题	(218)
上机练习	(226)
第六章 结构、联合和枚举	(227)
6.1 结构的概念	(227)
6.2 结构的应用	(236)
6.3 联合	(246)
6.4 枚举	(252)
本章小结	(257)
思考题	(259)

练习题	(260)
上机练习	(267)
第七章 文件	(268)
7.1 文件	(268)
7.2 标准文件的操作	(270)
7.3 一般文件的操作	(281)
本章小结	(298)
思考题	(300)
练习题	(301)
上机练习	(308)
第八章 特点和应用	(309)
8.1 怎样学好用好 C 语言	(309)
8.2 如何继续学好 C 语言	(317)
8.3 最后的例题	(320)
附 录	(325)
附录 A Turbo C 2.0 版编译系统使用方法简介	(325)
附录 B ASCII 编码表	(336)
附录 C Turbo C 常用库函数	(337)

第一章 实 例

了解C语言先从一个用C语言编写的程序实例开始。从对这个实例的分析来了解C语言程序结构和书写程序的格式。总之，通过一个实例了解C语言程序的外貌。

1.1 一个例子

用C语言编写一个程序，实现如下功能：已知两个正整数 m 和 n ，求 m 的 n 次方的值是多少？并要连续操作若干次。

按上述要求编程如下：

```
#define N 2
main ( )
{
    int m, n;
    system (" cls");
    printf("%40s\n", " * * * * *");
    printf("%40s\n", "求 m 的 n 次方");
    printf("%40s\n", " * * * * *");
    printf ("%40s", " 请输入 m 和 n 的值:");
    while (scanf("%d%d", &m, &n) == N)
    {
        printf (" \n\n");
        if (m<0&& n<0)
            exit ();
    }
}
```

```

printf ("%30s 当 m=%d, n=%d 时\n", " ", m, n);
printf ("%30s m 的 n 次方等于 %d\n\n", " ",
        power (m, n));
}
}
power (x, y)
int x, y;
{
    int i, p=1;
    for (i=0; i<y; i++)
        p *= x;
    return (p);
}

```

1.2 程序分析

一、执行该程序的输出结果

执行该程序首先在屏幕上显示如下信息：

```
*****
```

求 m 的 n 次方

```
*****
```

请输入 m 和 n 的值：

这时，在等待着键盘输入数值，当键入如下信息：

3 4↵

其中，3 和 4 是要赋给 m 和 n 的值， 表示空格，作为 3 和 4 的分隔符，↵表示回车键。

屏幕上接着显示如下信息：

当 $m=3$, $n=4$ 时
 m 的 n 次方等于 81

接着, 又出现如下信息:

请输入 m 和 n 的值:

当又键入两个数值后, 又显示出 m 的 n 次方的值。接着, 又让输入 m 和 n 的值, ……就这样反复地做下去, 直到当让输入 m 和 n 的值的, 用两个回车键来回答时为止, 这时便结束该程序的执行。

二、程序的简单分析

该程序作为一个文件存放在磁盘上。该文件有两个函数, 一个是主函数 `main()`, 另一个是被主函数调用的函数 `power()`。每个函数有各自的函数体用一对花括号 `{ }` 括起。

函数体中由若干个语句组成, 每个语句以分号“;”作为结束符。在诸多的语句中, 有说明语句, 用来说明变量的类型, 如在 `main()` 函数中, `int m, n;` 就是一个说明语句, 说明变量 m 和 n 是整型数。还有在 `power()` 函数中, `int i, p=1;` 也是一个说明语句, 说明变量 i 和 p 是整型数, 并且给变量 p 赋了初值, 即让变量 p 获得的值为 1。函数体中其他语句都是执行语句, 这里有表达式语句、循环语句、分支语句、返回语句等等。这里许多语句都是函数调用, 例如: 主函数中, `system()`, `printf()`, `exit()`, `scanf()` 和 `power()` 等都是函数调用。所谓函数调用是指在一个函数体中, 转去执行被调用的函数的函数体, 执行完毕后再返回到这个函数体来继续向下执行。在被调用的函数中, 有些是系统已定义好的函数, 如在该例中的 `system()`, `printf()`, `exit`

() 和 scanf()，有些是用户自己定义的函数，如该例中的 power() 函数，它的定义就写在主函数 main() 的后边。系统提供的函数用户自然不用定义，只需调用它就可以了，可见系统定义的函数越多，用户使用起来就越方便。

每个语句是由若干个单词按一定规则组成的。例如，在 power() 函数中，循环语句 for 的循环体 $p * = x$ ；是一个赋值表达式语句，它等价于 $p = p * x$ ；即是将变量 p 的值与变量 x 的值相乘，将乘积赋给变量 p，那么 p 的值便是 p 与 x 的相乘积。可见，该语句由变量和运算符组成，其中，p，x 是变量，*，= 都是运算符。而它们又都属于 C 语言的单词。

C 语言的单词很多，它们是由字符组成的，所以，字符是组成单词的最小元素。

综上所述，不难看出一个 C 语言编写的程序是这样构成的：

程序 → 文件 → 函数 → 语句 → 单词 → 字符。

这就是说，C 语言的程序是由一个或多个文件组成的；一个文件又是由一个或多个函数组成；函数是组成程序的基本单位；一个函数又是由若干个语句构成函数体的；一个语句是由若干个单词组成的；单词是由字符组成的；字符是组成 C 语言程序的最小元素。

本书将从字符讲起，先讲字符集与单词，再讲语句，再讲函数。

这里需要记住，在一个单文件程序的若干个函数中必须有一个名为 main() 的主函数，并且只能有一个。而在一个多文件程序中，也只允许有一个主函数。

最后，我们看到：C 语言的程序无论是多少个文件组成的，它都是一个函数串。即 C 语言的程序是由若干个相互间

又独立又有联系的函数组成的。

三、程序中有待解决的问题

上述分析只能达到对 C 语言的程序有点初步的了解，或者说熟悉一下程序的概貌。至于该程序的输出结果为什么是所显示的那样，程序中每条语句所起的作用是什么？每个单词的具体含义又是什么？这些问题有待于后面详细描述，这也就是本书要讨论清楚的问题。

再看一遍程序就会发现这里有许多要解决的问题。程序开始出现的 `#define N 2`，这是一条宏定义的预处理语句，用这种方法定义符号常量 `N` 等于 2。什么是预处理语句？为什么要用预处理语句？有哪些常用的预处理语句呢？有待解决。程序中出现了主函数 `main()`，`main` 是主函数的名字，`()` 里应放函数参数，该主函数没有参数，是不是所有主函数都没有参数呢？主函数的参数是什么？做什么用？有待解决。与该主函数并列出现的还有一个 `power(x, y)` 函数，该函数名为 `power`，它具有两个参数 `x` 和 `y`，这两个参数都是整型数 `int`。接下去一对 `{ }` 给出了该函数的函数体。那么，这个程序是由两个并列出现的函数组成的。函数应该怎样定义？函数定义后在调用之前是否要说明？怎样说明？函数之间有何关系？函数间的信息如何传送？函数的调用有何特点？这些问题有待解决。再看函数体内，包含有许多条语句，有说明语句和执行语句，而说明语句都放在执行语句的前边。说明语句是用来说明或定义变量的，例如 `int m, n;` 和 `int i, p=1;`；说明 `m, n, i, p` 都是 `int` 型变量，`int` 是类型说明符，`m, n, i, p` 为变量名，`p=1` 在说明语句中表示给变量 `p` 赋初值为 1。其他变量 `m, n, i` 只是定义了，但没有给赋初值，此时，这些变量中的