

青少年微机应用自学读本



C 语言 编程入门

李梅 姚沂 编著

化学工业出版社

卷首语

时代考验着年轻人，年轻人创造着时代。即将步入21世纪的年轻人可谓是最最忙碌，他们先要用知识武装自己，还要弥补自己的实践，同时还要证明一下自己的能力，时不时还要为这个小小的地球村落增加一些新的点子。

我们这个创作组是一群年轻的教师，学习和工作的方法可以用二个字概括——疯狂。这恐怕和大家没有什么不同，与大家有些不同的话，就是我们教师的职业习惯——笔记和积累，正是这一笔财富，使我们有胆量步入书籍出版这一神圣的殿堂。

自始至终，我们都以创作艺术品的虔敬心情，精心雕刻各自的第一本书籍。以哲学观点看，这里所呈现给大家的，不过是一些自我概念的表达，但是，我们要传达给大家的却是我们心灵的合拍，因为，我们要一同去创造21世纪。

《青少年微机应用自学读本》

创作组

目 录

第一章 推开 C 语言之门	1
第一节 关于机器语言.....	2
第二节 什么是 C 语言.....	2
第三节 C 语言的特点.....	3
第二章 初步接触 C 程序	5
第一节 一个小程序.....	6
第二节 运行这个程序.....	6
一、进入 Turbo C 集成环境.....	6
二、编辑源文件.....	7
三、编译源程序.....	7
四、将源程序存入磁盘.....	8
五、执行程序.....	8
六、退出 TC.....	8
第三章 逐步学习基本概念	9
第一节 你将遇到的基础知识.....	10
一、简单介绍 C 程序.....	10
二、基础知识.....	11
第二节 常量和变量.....	13
一、常量.....	13
二、变量.....	16
第三节 运算符简介.....	19
第四节 算术运算符与算术表达式.....	20
一、基本的算术运算符.....	20
二、算术表达式和运算符的优先级.....	20
三、自增、自减运算符.....	22
第五节 赋值运算符和赋值表达式.....	22
一、赋值运算符.....	22
二、复合的赋值运算符.....	23
三、赋值表达式.....	23
第六节 逗号运算符和逗号表达式.....	24
习题一.....	25
第四章 编写最简单的程序	27
第一节 C 语句概述.....	28
第二节 C 程序的基本元素.....	29

第三节 变量赋初值.....	30
第四节 输出数据.....	30
一、putchar 函数(字符输出函数).....	30
二、printf 函数(格式输出函数).....	32
第五节 输入数据(从键盘).....	38
一、getchar 函数(字符输入函数).....	38
二、scanf 函数(格式输入函数).....	39
第六节 顺序结构程序.....	41
一、顺序结构.....	42
二、顺序结构编程练习.....	42
习题二.....	46
第五章 判断与选择.....	47
第一节 引言.....	48
第二节 关系运算符和关系表达式.....	48
一、关系运算符及其优先级.....	48
二、关系表达式.....	49
第三节 逻辑运算符和逻辑表达式.....	50
一、三种逻辑运算.....	50
二、逻辑运算符及其优先级.....	51
三、逻辑表达式.....	52
四、用逻辑表达式表示条件.....	52
第四节 if 语句.....	53
一、程序的选择结构.....	53
二、if 语句的格式.....	54
三、if 语句的简单应用.....	55
第五节 嵌套的 if 语句.....	59
一、if 语句的嵌套.....	59
二、简单应用.....	60
第六节 条件运算符.....	61
一、条件运算符和条件表达式.....	61
二、说明.....	62
第七节 程序练习.....	63
习题三.....	66
第六章 循环.....	67
第一节 概述.....	68
第二节 循环结构.....	68
一、两种不同的描述方法.....	68
二、两种循环结构.....	69

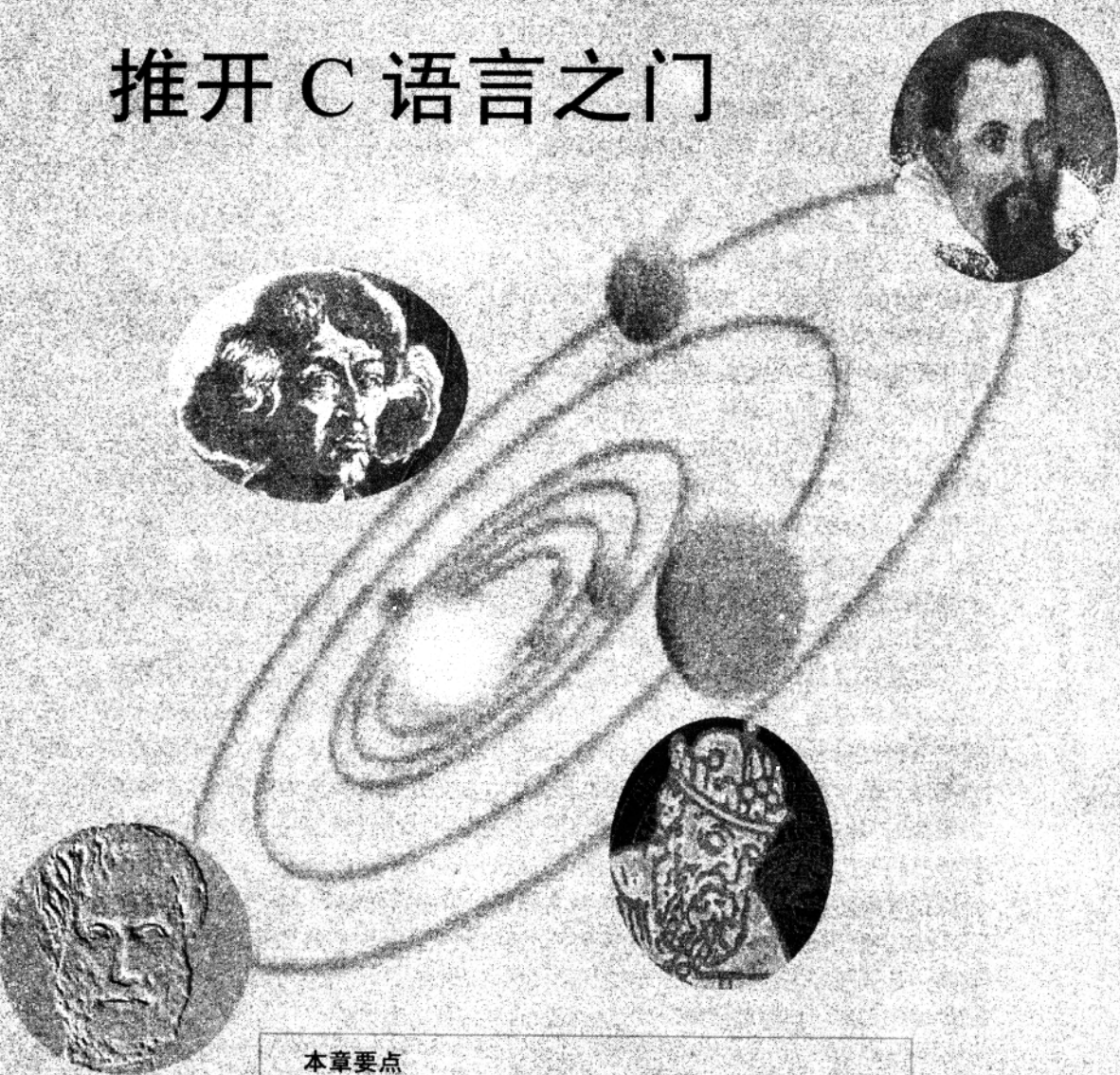
三、两种循环结构的区别	70
第三节 while 语句	70
一、while 语句的格式及功能	70
二、应用举例	71
第四节 do-while 语句	76
一、do-while 语句的格式及功能	76
二、应用举例	77
第五节 for 语句	80
一、for 语句的格式及功能	80
二、for 语句的特点	82
三、应用举例	83
第六节 三种循环的比较	85
第七节 循环的嵌套	85
一、循环的嵌套	85
二、应用举例	87
第八节 break 语句和 continue 语句	88
一、break 语句	88
二、continue 语句	88
三、break 和 continue 的区别	89
四、应用举例	89
第九节 程序举例	91
习题四	93
第七章 数组	95
第一节 引言	96
第二节 数组的概念	96
一、什么是数组	96
二、数组的维度	97
第三节 一维数组的定义和引用	97
一、一维数组的定义	97
二、一维数组元素的引用	97
三、一维数组的初始化	98
四、一维数组程序举例	99
第四节 一维数组的应用	101
一、排序	101
二、其他	105
第五节 二维数组的定义和引用	107
一、二维数组的定义	107
二、二维数组的引用	108

三、二维数组的初始化	108
四、二维数组应用举例	110
第六节 字符数组	113
一、定义字符数组	113
二、字符数组的初始化	113
三、字符数组的引用	114
四、字符串和字符串结束标志	114
五、字符数组的输入输出	116
六、常用字符串处理函数	117
习题五	120
第八章 结构体	121
第一节 概述	122
第二节 定义结构体类型	122
第三节 定义结构体类型变量	122
一、定义结构体类型变量的方法	122
二、关于结构体类型的几点说明	124
第四节 结构体类型变量的引用	125
一、结构体类型变量的引用	125
二、结构体变量的引用规则	126
第五节 结构体变量的初始化	126
第六节 结构体数组	129
一、结构体数组的定义	129
二、结构体数组的初始化	130
三、应用举例	130
习题六	133
第九章 函数	135
第一节 概述	136
第二节 函数定义	136
第三节 函数的分类	138
一、从用户使用角度分类	138
二、从函数形式分类	138
第四节 函数参数和函数的值	139
一、形式参数与实际参数	139
二、函数的返回值	140
第五节 函数的调用	141
一、函数调用的一般形式	142
二、函数调用的方式	142
三、对被调用函数的说明	142

四、函数应用举例	143
第六节 函数的嵌套调用	144
第七节 函数的递归调用	145
习题七	146
附录一 名词解释	147
附录二 常见错误分析	149

第一章

推开 C 语言之门



本章要点

计算机语言概述
C 语言的出现及特点

重要概念

指令 程序 机器语言 高级语言
源程序 目标程序 翻译程序 编译程序
编译 可移植性

第一节 关于机器语言

生活在地球上的人类，每个民族都拥有自己民族特色的语言——人类的自然语言，人们通过它相互沟通交流。计算机语言则是人类与计算机进行沟通的语言，我们可以用计算机语言表达出我们希望计算机执行的任务，让它快速而准确地完成。而这个用计算机语言所表达出来的、让计算机去解决问题的“文章”就是计算机程序。它安排、指定了计算机解决问题所采取的方法、步骤，并给出解决后的结果(实际上人们最关心、最需要得到的就是结果)。

在电子计算机问世的初期，人们对计算机发出的指令是用一串串数字“0”和“1”组成的二进制代码表示。例如在某个机器中，010000101010 是一条取数指令，前面四位“0100”表示让计算机执行“取数”的动作，后面八位“00101010”表示要取的数所在的地址。不同的机器有不同的指令格式，指令条数也不一样。由这些二进制数表示出的各种指令就是机器语言。很显然用机器语言编写程序是一件很令人头疼的事，因为机器语言难学、难记、难写，出了错误难于查错、纠错。

后来，经过计算机专家的研究，产生了高级程序设计语言(简称高级语言)，如 FORTRAN、COBOL、BASIC、C 语言等。高级语言是与计算机具体特性无关的程序语言，它使我们能以更接近于人类的自然语言或数学符号的形式编写程序、如用+表示相加，用 if 表示判断条件是否成立(if 意义为“如果”)。这样，用高级语言编写的程序能很好地反映人们求解一个问题的思路，它容易学习、易于记忆、易于书写、易于阅读，也容易修改。因此，现在编制程序通常都使用高级语言。

可是计算机只识别自己的机器语言，哪怕是贴近于人类自然语言的高级语言，计算机也是“读不懂、看不明”的，它又怎能去执行用高级语言编写的程序呢？当你要与一个外国朋友交谈而又不理解他所讲的语言时，你需要的是什么呢？是翻译人员。有了翻译，你们就可以交谈了。要想让计算机能识文断字、读懂高级语言编写的程序，就需要把它“翻译”成计算机的内部语言——机器语言，这样计算机就可以按照这一系列被“翻译”成的机器语言，依次执行。在这里，用高级语言编写的程序称之为源程序(意为计算机将要执行的指令的来源)，被“翻译”成的机器语言程序被称为目标程序。担当翻译人员角色的被称为编译程序，它是由计算机专家设计出来的一种有翻译功能的程序。不同的高级语言有各自专门的编译程序。把源程序翻译成目标程序的过程称之为编译。

第二节 什么是 C 语言

通过前节所述，现在你已有这样一个印象：C 语言是高级程序设计语言的一种。

C 语言是 70 年代初由美国电话电报公司(AT&T)所属的贝尔实验室的 D 里奇(Dennis Ritchie)为开发操作系统而独立设计并实现的。

1963 年英国的伦敦剑桥大学推出了 CPL(Combined Programing Language)语言, 1967 年英国剑桥大学的 M·里查德(Matin Richards)对 CPL 语言作了简化, 推出了 BCPL(Basic Combined Programing Language)语言。1970 年美国贝尔实验室的 K·汤普森(Ken Thompson)以 BCPL 语言为基础, 又作了进一步简化, 设计出了很简单的 B 语言(取 BCPL 的第一个字母)。1972 至 1973 年 D·里奇在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母), 这样 C 语言诞生并且有了自己的名字——C。

后来, 随着 UNIX 的日益广泛应用, C 语言也迅速得到推广(二者在发展过程中相辅相成)和发展, 先后推出了许多版本。美国标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充, 制定了 C 的标准——ANSI C(即标准 C)。目前广泛流行的各种版本 C 语言编译系统, 都是在 ANSI C 制定的标准下开发而成的, 它们基本部分相同, 也有一些不同。在微机上使用的有 Microsoft C、Turbo C、Quick C 等。本书中的例题均是在 Turbo C 上运行的。

第三节 C 语言的特点

C 语言有以下几个特点。

(1) 简洁

用 C 语言编写的程序往往短小精悍, 源代码的行数不多, 但功能却很强, 使用灵活, 易于学习和应用。

(2) 功能强

C 语言是一种通用的语言, 在不同型号计算机上都可运行, 适用范围广, 如科学计算、商业处理、软件开发等方面。

(3) 效率高

许多试验表明, 针对同一个问题, 用 C 语言编写的程序在执行时速度比用其他高级语言(如 FORTRAN、BASIC 等)编写的程序快很多。

(4) 可移植性好

可移植性是指编写的程序不用作太多的改动就可从一种机移到另一种计算机上运行, C 语言在这方面有专长, 从而有较强的生命力。

另外, C 语言是面向结构的语言, 一个大问题可由若干小部分组成, 像搭积木那样。掌握了 C 语言的基本知识, 对以后进一步提高程序设计能力, 以至实际开发软件, 都是很有益处的。

C 语言有很多优点, 但它也存在某些缺点。例如运算符(表示要进行运算的符号)较多、某些运算符优先级与习惯不完全一致等。尽管如此, C 语言仍以其众多优点博得计算机界的交口称赞, 备受人们的青睐, 成为高级语言的佼佼者。它就像一个可爱的小精灵, 活跃于世界各地。

第二章

初步接触 C 程序



本章要点

一个 C 语言程序的概貌
如何上机运行 C 程序

重要概念

编译 源程序

看到 C 语言是这么令人喜爱，你是不是也跃跃欲试，想亲自体验一下用 C 语言编写程序的快乐？别急，在这章里我们先了解一下 C 程序的格式以及如何编辑、编译、连接、运行一个 C 语言程序。

第一节 一个小程序

接触过高级程序设计语言的人都知道，对于不同的语言有其各自要求的语句格式、书写程序的约定。这里有一个非常简单的小程序：

【例 2-1】hello.c

```
main()
{
printf("Hello world!\n");
printf("I am here.\n");
}
```

如果我们把计算机输出设备指定为显示器，那么这个程序被计算机执行后，在屏幕上显示如下内容：

```
Hello world!
I am here.
```

这就是程序执行后的运行结果。

第一程序行中的 main() 是主函数名，它在任何一个 C 程序中必须出现，且只能出现一次。

第二程序行和第五程序行的一对花括号用于括起函数体，被括在内的是构成函数体的语句，要求每个语句的最后一个符号是分号“；”。

由此总结出，简单 C 程序的框架：

```
main()  主函数名
{
.....  函数体
}
```

第二节 运行这个程序

本书以 Turbo C 2.0 为例。

美国 Borland 公司推出的 Turbo C 是一个集程序编辑、编译、连接、调试于一体的 C 语言程序开发软件，具有速度快、效率高、功能强等优点，使用非常方便，你可以在 Turbo C 集成环境下进行全屏幕编辑，利用窗口功能进行编译、连接、调试、运行等工作。

一、进入 Turbo C 集成环境

将 Turbo C(以下简称 TC)装入机器硬盘后，在 DOS 操作系统提示符下进入 TC 子目录，键入

TC

屏幕上出现 TC 的初始画面，然后进入主菜单，主菜单内容如下：

```
FILE EDIT COMPLIER PROJECT OPTIONS DEBUG
```

用键盘上的光标移动键“←”和“→”可移动光标选择主菜单中各项操作(在集成环境的任何状态下，只要按“F10”即可进入选择主菜单中命令的状态)。

二、编辑源文件

在 TC 集成环境下，若想运行一个你编写的 C 语言程序，必须给这个程序建立一个源程序文件。

(1) 选择主菜单中最左端的“FILE”命令(即将光标移置“FILE”处，按回车键)，“FILE”下面会出现一个窗口

```
Load          F3
Pick          Alt-F3
New
Save          F2
Write to
Directory
Change dir
Os shell
Quit          Alt-X
```

它是一个子菜单，提供多项选择，可用光标移动键“↑”和“↓”选择其中各项内容。

(2) 选择“Load”命令，屏幕出现一个小窗口

```
Load File Name
Hello.c
```

此时输入要建立的源程序文件名“Hello.c”(文件名的扩展名部分可以省略，TC 在存盘时会自动加上扩展名“.c”)，如果这是你第一次以“Hello.c”为文件名建立源文件，则回车后进入编辑状态；若当前工作目录下存在“Hello.c”文件，则 TC 将调入该文件并显示在屏幕上，然后进入编辑状态。

(3) 输入(或修改)源程序



TC 编辑器的使用与其他流行的字符编辑器基本相同，若需要详细资料可按“F1”键查看帮助信息。

三、编译源程序

按“F9”键即可进行编译、连接，并在屏幕中央一个小窗口内显示有几处错误，按任意键后，该窗口消失，屏幕显示源程序，光标停留在出错处，屏幕下半部分显示出错所在行和出错原因，根据出错信息，检查并修改源程序，再重复上述编译过程，直至不出现错误为止。

四、将源程序存入磁盘

按“F10”键从主菜单选择“FILE”，然后选择子菜单中的“Save”选项即可存盘；或在编辑状态下直接按“F2”键可将正在编辑的文件存盘，存盘完毕自动回到编辑状态。

五、执行程序

按“F10”键，从主菜单中选择“RUN”，然后选择子菜单中的“Run”选项便可执行刚编译好的程序“Hello.c”。

按 Alt-F5 键，屏幕可切换到运行结果显示状态，再按任意键，回到显示源程序状态。

若发现运行结果不对，可按“F10”键，从主菜单中选择“EDIT”进入编辑状态修改源程序。修改完毕，再进行编译、运行，直到运行结果正确为止。

六、退出 TC

当你上机完毕后，应退出 TC 返回 DOS 系统再运行其他软件或关机休息。

退出 TC 时应从主菜单中选择“FILE”，从“FILE”子菜单中选择最后一项“Quit”即可退出 TC；或按“Alt-X”键直接退出。



在上机时，要养成及时将源文件存盘的习惯，免得自己的成果白白丢失。

比如：你正在输入一个一千行的源程序，输入到九百行时突然停电；你的程序已编译成功，但运行时出现死机，而你在输入过程中没有及时存盘。

上述步骤你上机试几次就能熟练操作，欲详细了解上机操作各个方面，可参阅 Turbo C 的使用手册。

第三章

逐步学习基本概念



本章要点

常量和变量
运算符和表达式
数据输入
数据输出

重要概念

常量 变量 关键字 位 字节

第一节 你将遇到的基础知识

一、简单介绍 C 程序

先看一个程序。

【例 3-1】

```

main()           ——主函数名
{
  int a,b,sum;   —— 变量定义部分
  a=5;
  b=9;
  sum=30+a+b;
  printf("sum is %d\n",sum);
  printf("OK!\n");
}
    
```

} 语句执行部分

这个程序的运行结果是：

```

sum is 44
OK!
    
```

看到这个运行结果，细心的你不难看出这个程序的功能，是计算出式子 $30+5+9$ 的值并将值输出。该程序的函数体内有两部分内容：变量定义部分和语句执行部分。（在程序中要使用的）变量必须先定义，才能在语句执行部分中使用。

程序中最后两行将计算结果输出并在屏幕上换行打印“OK！”。C语言没有用于输入、输出的语句，由C函数库中提供的库函数 `scanf`(输入)和 `printf`(输出)完成。

该程序可如下书写：

```

main()
{
  int a,b,sum;           /*定义变量 a,b,sum 为整型*/
  a=5;                  /*给变量 a 赋值 5*/
  b=9;                  /*给变量 b 赋值 9*/
  sum=30+a+b;          /*计算表达式值并赋给 sum*/
  printf("sum is %d\n",sum); /*输出 sum 值*/
  printf("OK!");        /*输出字符串 OK*/
}
    
```

用符号“/*”和“*/”括起的部分称为注释内容。在程序执行时注释内容不影响运行结果，也不起任何作用，它只是帮助编程者或他人能尽快地读懂、更好地理解程序。

总结：

- (1) 一个 C 程序由函数构成，必须包含一个主函数 `main()`，且只能有一个主函数。
- (2) 主函数由函数名 `main` 和函数体构成，函数体包括变量定义和语句执行两部分，由花括号括起。