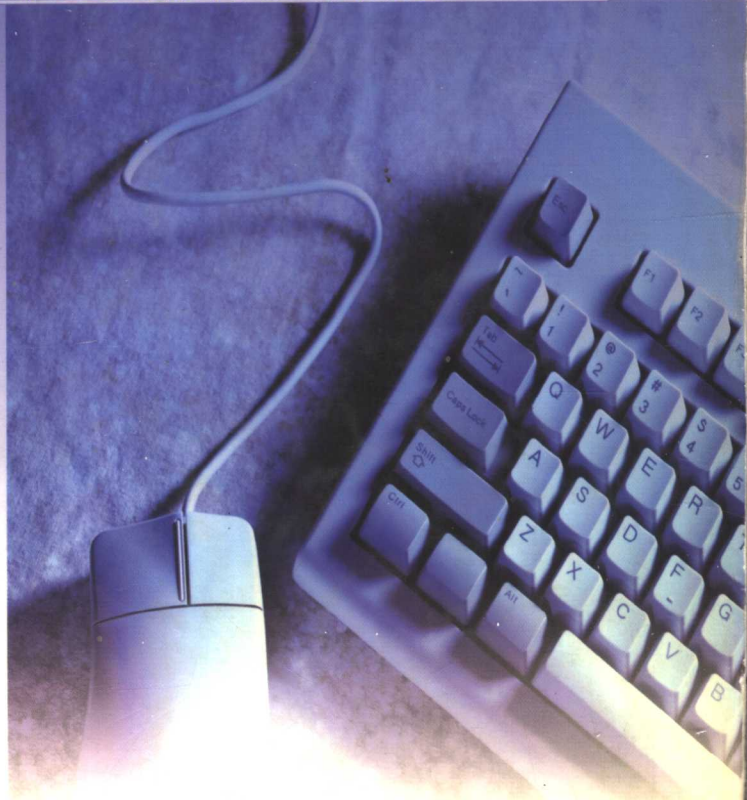


高等学校计算机基础教材精选



崔武子 赵重敏 李青 著

C 程序设计教程



清华大学出版社

高等学校计算机基础教育教材精选



C
程序

设计教程

崔武子
赵重敏
李青
著

清华大学出版社
北 京

内 容 简 介

本书是以 C 程序设计零起点读者作为主要对象的程序设计教程。

全书用例子组织所有的教学内容,整体内容编排独特、组织形式新颖:1. 将每章内容及习题,均分成必须掌握的基础部分和具有扩展性的提高部分,学时不足时,跳过提高部分不影响后续内容的学习;2. 针对学生“读程序容易,编程序难”的情况,书中凡涉及到算法的例子,在给出完整程序之前,都增设了编程点拨,有些算法还提供了多种解法;3. 将贯穿整个教学过程的一个实例分阶段给出了相应的程序,随着讲授内容的增多,逐步完善其程序,最终达到实现实例的全部功能;4. 分阶段通过具体例子介绍了调试程序的方法,以使读者及时体会不同阶段程序的调试过程;5. 习题形式新颖多样、由浅入深,而且提供单号习题的答案,方便学生自测和教师留作业;6. 配备了专门辅导与实训教材及内容生动的电子教案。

本书是高等院校 C 程序设计课程的教材,也可作为 C 语言自学者的参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

C 程序设计教程/崔武子,赵重敏,李青著. —北京:清华大学出版社,2003

(高等学校计算机基础教育教材精选)

ISBN 7-302-06718-X

I. C… II. ①崔… ②赵… ③李… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 045588 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

责任编辑: 谢 琛

印 刷 者: 北京四季青印刷厂

装 订 者: 北京国马印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 22 字数: 496 千字

版 次: 2003 年 7 月第 1 版 2004 年 4 月第 2 次印刷

书 号: ISBN 7-302-06718-X/TP·5010

印 数: 6001~8500

定 价: 28.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或(010)62795704。

出版说明

—— 高等学校计算机基础教育教材精选 ——

在教育部关于高等学校计算机基础教育三层次方案的指导下,我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践,全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验,取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快,目前我国的高等教育事业正面临着新的发展机遇,但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要,进一步推动我国高等学校计算机基础教育事业的发展,我们在全中国各高等学校精心挖掘和遴选了一批经过教学实践检验的优秀教学成果,编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次,面向各高校开设的计算机必修课、选修课以及与各类专业相结合的计算机课程。

为了保证出版质量,同时更好地适应教学需求,本套教材将采取开放的体系和滚动出版的方式(即成熟一本、出版一本,并保持不断更新),坚持宁缺勿滥的原则,力求反映我国高等学校计算机基础教育的最新成果,使本套丛书无论在技术质量上还是文字质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作,在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度,以适应高校计算机基础教育课程层次化、多样化的趋势,从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来,把自己的教学成果与全国的同行们分享;同时也欢迎广大读者对本套教材提出宝贵意见,以便我们改进工作,为读者提供更好的服务。

我们的电子邮件地址是: jiaoh@tup.tsinghua.edu.cn(E-mail);联系人:焦虹。

清华大学出版社

2001年8月

前言

C 程序设计教程

本书是以 C 程序设计零起点读者作为主要对象编写的程序设计教程。本书的内容编排独特、组织形式新颖,能使读者在较短的时间内掌握 C 程序设计的精华。本书是高等院校 C 程序设计课程的教材,也可作为自学者的指导书。

本书特点:

(1) **每章内容分成基础部分和提高部分。**考虑到 C 的语法现象众多、初学者往往难于接受,书中将每章的内容分成了基础和提高两个部分。将常识性的、基础类的、必须掌握的内容放在基础部分中;将具有扩展性的、提高性的、专业性的内容安排在提高部分中。通过基础部分的学习,能够掌握最基本的语法,初步建立程序设计的思维方式和编写一般的程序,同时可培养学生的学习兴趣。即使因学时不足跳过提高部分,也不至于影响后续内容的学习。

(2) **所有教学内容用例子组织。**在基本遵循 C 语言教学体系的情况下,将所有教学内容用例子组织。即根据要介绍的内容精心编写相应的例子,将大量的、正确的、规范的程序介绍给学生,在讲解例子的过程中,使学生学习语法、了解概念、掌握算法。做到在解决实际问题中讲授语法,而不是为了教语法而举例子。

(3) **涉及算法的例子增设编程点拨,部分算法提供多种解法。**针对学生“读程序容易,编程序难”的情况,书中凡涉及算法的例子,在给出其完整的程序之前,都增设了编程点拨,有些算法还提供了多种解法。

(4) **分阶段介绍程序调试方法。**为了培养学生调试程序、排除错误的能力,教材中分阶段通过具体例子介绍了调试程序的方法。

(5) **讲授指针和函数时不涉及新算法。**指针和函数是 C 语言中的重点和难点,为了使学生能够顺利接受新概念,本教材将有关算法内容尽量安排在这两章之前,避免学生在接受指针和函数概念的同时,又要理解新算法。这样做不仅容易突破难点,而且有利于巩固已学过的知识。

(6) **用实例贯穿整个教学过程。**为了使学生尽早接触应用程序的编写过程,在提高部分中,提供了贯穿于整个教学过程的一个实例,并随着讲授内容的增多,逐步补充和完善其程序的功能。

(7) **习题形式新颖,提供单号习题答案。**与教材内容相对应,各章习题也分为基础和提高两部分。为了逐步提高学生的编程能力,还精心编写了形式新颖的习题。书中单号习题提供参考答案,以方便学生自测和教师布置作业。

(8) 配备含有电子教案等内容的光盘。该光盘包括电子教案、全书所有例子的源程序以及实例的可执行文件。为了减轻教师备课的负担,本教材将基础部分中的所有内容和提高部分的部分内容以及完整实例的演示,制作成生动的电子教案,通过电子教案可使读者尽早了解本课程的基本目标。

(9) 配套出版《C 程序设计辅导与实训》。该书内容既兼顾了随教学进度安排的阶段实训,也考虑到全部教学内容完成后的集中实训。实训部分以本教材中的实例作为样例;辅导部分提供了大量例题的解析、训练题和参考答案,因此是本教材的最佳教学辅助资料。此书也将随后由清华大学出版社出版。

使用建议:

(1) 必学基础部分。基础部分是学生必须掌握的知识,但在教学过程中教师可将部分例子留给学生自学。

(2) 选学提高部分。书中的提高部分是为了帮助读者更上一层楼,教师可以根据实际情况,选择其中部分内容进行介绍(标有“*”的例子有一定难度)。为了提高学生的上机编程和调试能力,建议教师指导学生学习其中实例程序。

(3) 单、双号习题成对做。单号习题提供参考答案,双号习题则在类型上与前一单号习题相同,侧重点也接近。基础部分中提供的习题都是最基本的,题量也不多,建议读者全部做完,提高部分中的习题可根据情况选做(标有“*”的习题有一定难度)。

本书由崔武子主编,赵重敏和李青参加了部分内容的编写。全书由崔武子统稿,鲍有文副教授审阅。在编写本书的过程中,得到谭浩强教授和高林教授的热情关怀和指导。齐华山等老师给本书和电子教案提出了许多宝贵意见,在此一并表示真挚的感谢。

限于作者水平,书中难免有错误和疏漏之处,恳请读者批评和指正。

作 者

2003 年 2 月

目录

C 程序设计教程

第 1 章 C 语言基础知识	1
1.1 C 语言概述	1
1.1.1 什么是 C 语言	1
1.1.2 C 程序形式和程序执行过程	2
1.2 简单 C 程序与上机步骤	3
1.2.1 简单 C 程序	3
1.2.2 上机步骤	5
1.3 数据类型	7
1.4 常量与变量	8
1.4.1 常量与变量的概念	8
1.4.2 整型常量与变量	10
1.4.3 实型常量与变量	11
1.4.4 字符型常量与变量	13
1.5 运算符和表达式	15
1.5.1 算术运算符和表达式	15
1.5.2 赋值运算符和表达式	17
1.5.3 逗号运算符和表达式	19
1.5.4 自加、自减运算符	20
1.6 提高部分	21
1.6.1 Turbo C 集成环境	21
1.6.2 赋值运算符的进一步讨论	24
1.6.3 自加、自减运算符的进一步讨论	25
1.6.4 整型数据类型的进一步讨论	27
习题 1	29
1_1 基础部分习题	29
1_2 提高部分习题	31
第 2 章 顺序结构程序设计	32
2.1 结构化程序设计的基本结构	32

2.1.1	语句的概念	32
2.1.2	三种基本结构	33
2.2	赋值语句	35
2.3	输入输出语句	36
2.3.1	格式输入输出函数	36
2.3.2	字符输入输出函数	39
2.4	提高部分	40
2.4.1	输入输出函数的进一步讨论	40
2.4.2	实例的部分程序	44
习题 2		46
2_1	基础部分习题	46
2_2	提高部分习题	46
第 3 章	分支结构程序设计	48
3.1	关系运算符和关系表达式	48
3.1.1	关系运算符	48
3.1.2	关系表达式	49
3.2	逻辑运算符和逻辑表达式	50
3.2.1	逻辑运算符	50
3.2.2	逻辑表达式	50
3.3	if 语句	52
3.3.1	if 语句的一般形式	52
3.3.2	if 语句的嵌套	60
3.4	switch 语句	64
3.5	提高部分	67
3.5.1	if 和 switch 语句的进一步讨论	67
3.5.2	条件运算符和表达式	75
3.5.3	实例的部分程序	76
习题 3		78
3_1	基础部分习题	78
3_2	提高部分习题	79
第 4 章	循环结构程序设计	82
4.1	for 语句	82
4.2	while 语句	91
4.3	do-while 语句	95
4.4	break 语句和 continue 语句	98
4.4.1	循环体中使用 break 语句	98

4.4.2	循环体中使用 continue 语句	100
4.5	循环语句的嵌套	101
4.6	提高部分	104
4.6.1	for 语句的应用	104
4.6.2	三种循环的对比	108
4.6.3	goto 语句以及用 goto 语句构成的循环	112
4.6.4	实例的部分程序	113
习题 4		116
4_1	基础部分习题	116
4_2	提高部分习题	119
第 5 章	数组	122
5.1	一维数组	122
5.1.1	一维数组的定义和引用	122
5.1.2	一维数组的初始化	125
5.1.3	一维数组程序举例	128
5.2	字符串	140
5.2.1	字符串的概念和字符串的输入输出	140
5.2.2	字符串处理函数和实现相应功能的自编程序	143
5.3	二维数组	147
5.4	提高部分	150
5.4.1	二维数组的程序举例	150
5.4.2	实例的部分程序	154
习题 5		157
5_1	基础部分习题	157
5_2	提高部分习题	158
第 6 章	指针	159
6.1	变量的地址和指针变量的概念	159
6.2	指针变量的定义和引用	160
6.3	指针和一维数组	163
6.3.1	使指针变量指向一维数组	163
6.3.2	对指针的算术运算	164
6.4	指针和字符串	168
6.4.1	通过字符数组名引用字符串	168
6.4.2	通过指针变量引用字符串	169
6.5	提高部分	171
6.5.1	指针的进一步讨论	171

6.5.2	指针和二维数组	173
6.5.3	实例的部分程序	177
习题 6		186
6_1	基础部分习题	186
6_2	提高部分习题	188
第 7 章	函数	189
7.1	函数的引例	190
7.2	函数的定义与调用	192
7.2.1	函数的定义	192
7.2.2	函数的调用	193
7.2.3	函数的调用过程	198
7.2.4	函数的返回值	201
7.2.5	被调函数的原型说明	205
7.3	函数的嵌套调用	206
7.4	数组做实参	207
7.4.1	数组元素做实参	207
7.4.2	一维数组名做实参	208
7.4.3	二维数组名做实参	211
7.5	变量的存储类别	213
7.5.1	内部变量和外部变量	213
7.5.2	动态存储变量和静态存储变量	215
7.6	提高部分	216
7.6.1	函数的递归调用	216
7.6.2	带参数的 main 函数	219
7.6.3	指向函数的指针	224
7.6.4	变量存储类别的进一步讨论	226
7.6.5	预处理命令	228
7.6.6	实例的部分程序	232
习题 7		236
7_1	基础部分习题	236
7_2	提高部分习题	238
第 8 章	结构体和其他构造类型	241
8.1	结构体类型变量的定义和使用	241
8.1.1	结构体类型的概念和声明	241
8.1.2	结构体类型变量的定义和使用	242
8.2	结构体和函数调用	249



8.3 提高部分	252
8.3.1 结构体的进一步讨论	252
8.3.2 链表	256
8.3.3 共用体	264
8.3.4 枚举类型	266
8.3.5 实例的部分程序	268
习题 8	270
8_1 基础部分习题	270
8_2 提高部分习题	272
第 9 章 位运算与位段	274
9.1 移位运算符	274
9.2 按位与、或、异或运算符	275
9.3 按位取反运算符	277
9.4 提高部分	277
9.4.1 位运算的进一步讨论	277
9.4.2 位段	279
习题 9	280
9_1 基础部分习题	280
9_2 提高部分习题	281
第 10 章 文件	282
10.1 文件的概述	282
10.2 文件的基本操作	283
10.3 提高部分	291
10.3.1 文件读写操作的进一步讨论	291
10.3.2 文件的定位操作	297
10.3.3 实例的部分程序	301
习题 10	304
10_1 基础部分习题	304
10_2 提高部分习题	305
附录 I C 语言关键字	306
附录 II 常用字符与 ASCII 代码对照表	307
附录 III 运算符的优先级和结合方向	308
附录 IV 常用 C 库函数	309
附录 V 常见错误、警告信息表	312
单号题参考答案	313
参考文献	333

本章将介绍的内容

基础部分:

- C 程序的基本概念和上机步骤
- 各种数据类型的常量和变量
- 算术运算、赋值运算和逗号运算

提高部分:

- Turbo C 集成环境
- 再论赋值运算符和自加、自减运算符
- 再论整型数据类型

1.1 C 语言概述

1.1.1 什么是 C 语言

人和人之间交换信息需要借助于语言工具,人和计算机交换信息也同样要用语言工具。我们将后一种语言称为计算机语言。用计算机语言编写的代码称为程序。

随着计算机技术的发展,计算机语言逐步得到完善。最初使用的计算机语言是用二进制代码表达的语言——机器语言。后来采用与机器语言相对应的助记符表达的语言——汇编语言。虽然用这两种语言编写的程序执行效率高,但程序代码很长,又都依赖于具体的计算机,因此编码、调试、阅读程序都很困难,通用性也差。我们称这两种语言为低级语言。现在使用最广的计算机语言是高级语言——用更接近于人们自然语言的表达语言。高级语言独立于机器,编码相对短,可读性强。FORTRAN、QBASIC、Pascal、COBOL、C 等都是高级语言。用高级语言编写的程序叫做源程序。由于计算机只能识别 0 和 1,因此源程序必须通过编译和连接后,才能被计算机执行。

C 语言比其他高级语言功能更强,它具有高级语言的功能和低级语言的许多功能,即具有双重性,因此有的书把 C 语言称为中级语言。C 语言是由附录 I 中列出的 32 个关键

字再加上语法规则构成的。

要得到 C 程序的运行结果,首先将源程序输入到计算机内(在计算机上输入或修改源程序的过程叫做编辑),然后还要把源程序翻译成机器能识别的目标程序。如将编辑后的源文件保存为 e1.c,通过编译产生的与源文件相对应的目标程序则为 e1.obj;但是,目标程序不是可执行文件,不能直接运行,还要把目标程序和系统提供的库函数等连接起来产生可执行文件 e1.exe,这时才可以运行程序,并看到运行结果。C 程序的编辑、编译、连接、运行过程可用图 1.1 表示。

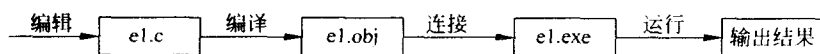


图 1.1 C 程序的编辑、编译、连接、运行过程

这一过程可以在不同的环境中进行,而本书中的所有例子是在 Turbo C 2.0 集成环境下运行通过的。Turbo C 2.0 是编辑、编译、连接和运行合为一体的集成环境,其操作简单易懂,在微机上得到广泛使用。在 1.6.1 中将介绍 Turbo C 2.0 的集成环境。

1.1.2 C 程序形式和程序执行过程

下面举一个 C 语言程序的完整例子,用此例说明 C 程序的一般形式和程序的执行过程。程序中的具体语法规则和其他细节将在后续章节中陆续介绍。

【例 1.1】 编写一个完整的 C 语言程序示例。

【解】 程序如下:

#include <stdio.h>		包含文件
#include <math.h>		
int mysum(int m, int n);	——	函数原型说明
main()	——	主函数首部
{ int a,b,x;		声明部分
double c,y,z;		
c=4.0;		
y=sqrt(c);		
a=10; b=20;		
x=mysum(a,b);		
z=x+y;		
printf("z=%lf\n",z);		
}		
int mysum(int m, int n)	——	mysum 函数首部
{ int k;	——	声明部分
k=m+n;		
return k;		语句部分
}		

程序的运行结果是：

$z=32.000000$

正如本例所示,C 语言程序是由函数构成的,任一 C 程序可以包含若干个函数,且至少包含一个函数,这个函数称为主函数。在 C 程序中主函数是必不可少的,而且程序都从主函数开始执行,主函数名必须是 main。例 1.1 中程序的执行过程如图 1.2 所示,程序按①到⑨的顺序执行。

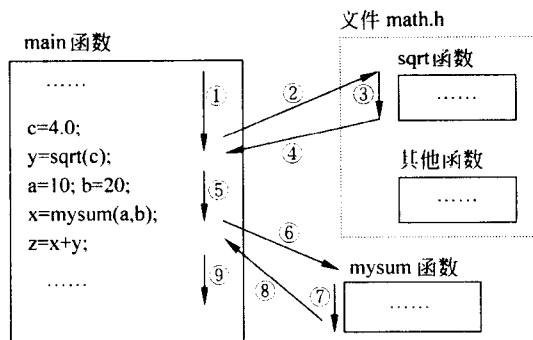


图 1.2 例 1.1 程序的执行过程

1.2 简单 C 程序与上机步骤

本节将给出几个最简单的 C 程序,通过这些例子,介绍 C 语言的基本概念,以及在 Turbo C 2.0 集成环境下的上机步骤。

1.2.1 简单 C 程序

【例 1.2】 编写在屏幕上显示一个句子“Let’s study the C language.”的程序。

【解】 程序如下：

```
#include <stdio.h>
main()
{
    printf("Let’s study the C language.");
}
```

运行结果如下：

Let’s study the C language.

程序说明：

(1) 程序中 main 是主函数名,每一个 C 程序都必须包含而且只能包含一个主函数。

用一对大括号“{}”扩起来的部分是函数体。本例函数体中只有一条语句“printf("Let's study the C language.");”。

(2) “printf("Let's study the C language.");”是输出语句,语句最后的“;”不能丢,此语句的作用是按原样输出双引号内的字符串 Let's study the C language. (详见 2.3.1 节)。

(3) C 语言中区分大小写,即:main 不能写成 Main,printf 也不能写成 Printf。若程序中有此类错误,则很难发现。C 语言中的所有关键字都用小写字母。

(4) 程序中的 #include <stdio.h> 称为命令行,有了此行,就可以成功地调用 C 语言标准库中提供的输入、输出函数,所以编写程序时,在程序的第一行都写此行。

【例 1.3】 编写输出两个句子“Let's study the C language.”和“It's interesting.”的程序。

【解】 程序如下:

```
#include <stdio.h>
main()
{
    printf("Let's study the C language.\n");      /* 输出字符串后换行 */
    printf("It's interesting.\n");
}
```

运行结果如下:

```
Let's study the C language.
It's interesting.
```

程序说明:

(1) 本程序的函数体包括两条输出语句,函数体可以包括任意多条语句。

(2) \n 是换行符,如果程序中去掉\n,输出形式则为:

```
Let's study the C language. It's interesting.
```

请思考: 程序中最后一个输出语句中的换行符(\n)的作用是什么?

(3) 程序中“/* 输出字符串后换行 */”是注释部分,对程序的运行无任何作用,注释的目的是方便阅读程序。注释可以出现在程序的任何位置。

小结:

(1) C 程序一定有且仅有一个主函数,主函数名必须是 main,其后的圆括号内可以是空的,但圆括号不能省略。

(2) 函数体中可以有多条语句,每条语句都以分号结束。

(3) 注释用/* */形式,注意,“/”和“*”之间不能有空格。为了提高程序的可读性,建议加上必要的注释。

(4) C 程序的书写格式比较自由。例如,一行内可以包括多条语句,一条语句也可以写在多行上,每行的内容可以从任何一列开始写。但是对于初学者建议每行写一条语句,而每行的语句根据需要适当缩进几列(参见例 4.20),这对读懂程序和调试程序很有帮



助。本书中的所有程序都按最常用的书写格式,请读者参考。

1.2.2 上机步骤

下面在 Turbo C 2.0 集成环境下,给出例 1.2 和例 1.3 的上机步骤。

先介绍例 1.2 的上机步骤。

第一步:在 C 盘 TC 目录下安装 Turbo C 2.0。如果已经安装,则跳过此步。

第二步:进入 Turbo C 集成环境。在 Windows 的“开始”菜单下选择“程序”中的“MS-DOS 方式”,将显示 DOS 提示符 C:\WINDOWS>,键入(在本书中用户输入的部分都用下划线表示)

```
C:\WINDOWS>cd \tc<回车>
```

```
C:\TC>tc<回车>
```

或键入

```
C:\WINDOWS>cd \<回车>
```

```
C:\>cd tc<回车>
```

```
C:\TC>tc<回车>
```

或直接键入

```
C:\WINDOWS>\tc\tc<回车>
```

都可进入 Turbo C 集成环境。如果屏幕中出现乱码,则单击屏幕上方的“中文”按钮。按 Alt 和 ENTER 组合键,可使屏幕变成最大,再按一次此组合键,可以恢复原屏幕大小(详见 1.6.1 节)。

第三步:如果显示本产品的版本信息,按 Esc 键取消信息,否则跳过此步。

第四步:编辑源程序。如果主菜单处于被激活状态,则按 Esc 键。在编辑区里输入例 1.2 中的程序,如果输入有错,修改之。

第五步:保存文件。因为上机时经常会发生预料不到的事情,一定要养成随时存盘的好习惯。按 F10 键,激活主菜单,在 File 菜单下选择 Write to,在出现的输入框内输入 d:\cwz\Exmpl_2.c(假设已在 D 盘根下建立 cwz 文件夹),如果只输入文件名,则将文件存放在当前盘当前目录下。文件名的默认扩展名为 c。File 菜单中的 Write to 命令一般用于第一次存文件或改名存盘,而 Save 命令则用于按原名存盘。

第六步:编译和连接程序。C 源程序通过编译和连接之后才能运行。按 F9 键可先将程序编译成目标程序 Exmpl_2.obj,再连接产生可执行文件 Exmpl_2.exe。如果编译或连接时没有错误则执行下一步,否则回到第四步。修改之后再行存盘操作,由于不是第一次存盘,选择 Save 命令,F2 是 Save 命令的热键。当需要编辑区和消息区之间切换,则按 F6 键。

第七步:运行程序。按 Ctrl 和 F9 组合键,系统就会运行 Exmpl_2.exe。如果运行时发现有错误,回到第四步,否则转到第八步。

第八步：查看运行结果。正常运行程序后，屏幕上仍显示 Turbo C 的窗口，为了查看运行结果，一定要切换到 DOS 屏幕。按 Alt 和 F5 组合键，Turbo C 集成环境将切换到 DOS 屏幕（黑色屏幕），并显示运行结果，即显示：

```
Let's study the C language.
```

按任意键，回到 Turbo C 环境，如果运行结果与预期的结果不相符，则重复第四至八步，直到结果正确为止。

在进行例 1.2 的上机操作后，不必退出 Turbo C 环境，可继续进行例 1.3 的上机操作，其步骤如下：

第一步：新建文件。选择 File 菜单下的 New 命令。

注意：在例 1.2 下方接着输入例 1.3 是错误的，因为系统认为这是一个 C 程序，而一个程序中不能有两个主函数。这是初学者容易出现的错误。

第二步：编辑源程序。输入例 1.3 中的程序，如果输入有错，修改之。

第三步：保存文件。在 File 菜单下选择 Save 或按 F2 键，第一次存盘也可以使用此命令。在出现的输入框内输入 d:\cwz\Exmpl_3.c。

第四步：编译、连接和运行程序。按 Ctrl 和 F9 组合键。系统先将程序编译成目标程序 Exmpl_3.obj，再连接产生可执行文件 Exmpl_3.exe，最后运行 Exmpl_3.exe，如果编译、连接或运行时没有错误则转到下一步，否则回到第二步。这种操作方式比较简便，它将编译、连接和运行过程统一进行，但系统不显示编译时的警告错误，而警告错误也是不容忽视的，因此建议读者今后尽量按例 1.2 中的第六、七步操作。

第五步：查看运行结果。按 Alt 和 F5 组合键，在 DOS 屏幕上将显示：

```
Let's study the C language.
```

```
It's interesting.
```

按任意键，回到 Turbo C 环境，如果运行结果与预期的结果不相符，则重复第二至五步，直到结果正确为止。

注意：第一次运行例 1.3，第一行的输出结果没有从第一列开始，这是因为在前次的运行程序中（例 1.2），输出字符串的最后没有“\n”，致使输出操作完成后，光标不能返回到屏幕的首列，例 1.3 的第一个输出就从光标当前停留位置开始。由于例 1.3 第二条输出字符串后面有“\n”，再次运行时，可以使光标返回到屏幕首列，以后的输出就会从第一列开始。

第六步：退出 Turbo C 环境。如果暂时不使用 Turbo C 环境，可以通过按 Alt 和 X 组合键退出。如果程序修改后没有存盘，系统将会给予存盘提示，否则直接退到 DOS 环境。

第七步：键入 exit，回到 Windows 界面。

说明：

如果在例 1.2 程序的输出语句尾部加“\n”，并在其后键入语句“printf(“It's interesting.\n”);”，再用 Write to 命令将文件 Exmpl_2.c 另存为 Exmpl_3.c，同样可以达到第一至三步的效果。

