



程序设计教程

主编 © 沈丽容 章春芳

C语言是一种强大、高效、优美的程序设计语言。
本书力求使学生掌握C语言基本语法和程序设计基础知识，
领会计算机编程思想，
掌握编程方法和技巧，
具备一定的程序设计和调试能力。

C PROGRAMMING TUTORIAL



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

C 程序设计教程

主 编 沈丽容 章春芳

东南大学出版社
· 南 京 ·

内 容 简 介

本书是根据教育部高等学校计算机科学与技术教学指导委员会提出的《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求》的有关要求编写的。

C 程序设计是一门实践性很强的课程,本书力求使学生掌握 C 语言基本语法和程序设计基础知识,领会计算机编程思想,掌握编程方法和技巧,具备一定的程序设计和调试能力,为今后学习和工作打下良好的基础。

本书内容结构规划合理,条理清晰,语法讲解细腻,突出重点、难点和易错点,案例分析透彻,课后习题丰富,既可作为高等学校各专业程序设计课程教材,又可以作为初学者自学用教材,还可作为等级考试辅导教材。

图书在版编目(CIP)数据

C 程序设计教程 / 沈丽容, 章春芳主编. — 南京:
东南大学出版社, 2015.8

ISBN 978-7-5641-5973-3

I. ①C… II. ①沈… ②章… III. ①C 语言-程
序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 189387 号

C 程序设计教程

出版发行 东南大学出版社
责任编辑 夏莉莉
出 版 人 江建中
社 址 南京市四牌楼 2 号
邮 编 210096

经 销 全国各地新华书店
印 刷 南京雄州印刷有限公司
开 本 787 mm × 1092 mm 1/16
印 张 22
字 数 475 千字
版 印 次 2015 年 8 月第 1 版 2015 年 8 月第 1 次印刷
书 号 ISBN 978-7-5641-5973-3
定 价 45.00 元

(凡因印装质量问题,请与我社读者服务部联系。电话:025-83791830)

前 言

随着信息技术的发展,计算机在社会中的应用可以说是无孔不入。20 世纪最重要的发明——计算机正在改变我们的学习、工作和生活方式。大家耳熟能详的 WPS、Office 办公软件,Windows、Mac OS、安卓操作系统,百度、淘宝等网站,都离不开编程。编程赋予了计算机多种多样的功能,使计算机成为人们的学习工具、工作助手和娱乐伙伴。如果没有编程,就没有软件产生;没有软件,计算机就没有这么神通广大。

面对当今极速发展的信息技术,社会对人才的培养提出了更高的要求,迫切 need 加强高等学校程序设计的教学工作。因此,当代大学生非常有必要了解计算机工作原理,掌握计算机编程思想和方法,提高用计算机解决问题的能力。通过学习 C 程序设计,学生不仅可以掌握 C 语言语法和编程技巧,而且可以提高逻辑思维和计算思维能力,可以更好地适应信息社会的需求。

本书由教学经验丰富的教师编写。沈丽容老师编写了第一、二、三、四、八章,章春芳老师编写了第五、六、七、九章。在本书的编写过程中参考了大量纸质和网上的文献资料,我们力求在参考文献中列全,对相关文献的作者,也在此表示衷心的感谢。

由于时间仓促和编者水平有限,书中欠妥和不足之处,恳请读者批评指正。

编者

2015 年 6 月

目 录

第一章 引言	1
1.1 学习编程有什么用	1
1.2 C语言的发展历程	1
1.3 C语言的特点	2
1.4 如何学习C语言	3
1.5 Turbo C 2.0 简介	3
1.6 Visual C ++6.0 简介	9
习题一	14
第二章 顺序结构程序设计	16
2.1 C源程序结构	16
2.2 最简单的C程序	17
2.3 整型数据	19
2.3.1 整型常量表示形式与定义	19
2.3.2 整型变量与整型常量的类型	19
2.3.3 整型变量定义	20
2.3.4 整型变量赋值	21
2.3.5 整型数据输入	24
2.3.6 整型数据输出	26
2.3.7 整型数据的运算	28
2.3.8 整型数据编程	30
2.4 实型数据	35
2.4.1 实型常量	36
2.4.2 实型变量	36
2.4.3 实型变量赋值	37
2.4.4 实型数据输入	37
2.4.5 实型数据输出	39
2.4.6 实型数据编程	40
2.5 字符型数据	43
2.5.1 字符常量	44
2.5.2 字符串常量	45
2.5.3 字符变量定义	47

2.5.4 字符变量赋值	47
2.5.5 字符型数据输入	48
2.5.6 字符型数据输出	48
2.5.7 字符型数据编程	49
2.6 各种类型数据之间的混合运算	51
2.6.1 自动类型转换	51
2.6.2 强制类型转换	53
2.7 C 语言的运算符、表达式和语句	54
2.8 逗号运算符和逗号表达式	55
习题二	56
第三章 选择结构程序设计	61
3.1 关系运算符和关系表达式	61
3.2 逻辑运算符和逻辑表达式	62
3.3 if 语句	65
3.3.1 最基本的 if 语句	65
3.3.2 if-else 语句	67
3.3.3 if-else if 语句	69
3.3.4 if 语句的嵌套	74
3.4 switch 语句	79
3.5 条件运算符和条件表达式	86
3.6 编程实战	88
习题三	94
第四章 循环结构程序设计	98
4.1 while 语句	98
4.2 do-while 语句	104
4.3 for 语句	109
4.4 goto 语句	113
4.5 循环嵌套	114
4.6 break 语句	115
4.7 continue 语句	117
4.8 编程实战	118
习题四	125
第五章 数组	130
5.1 一维数组的定义和应用	130
5.1.1 一维数组的定义	131
5.1.2 一维数组元素赋值	131
5.2 二维数组的定义和应用	138

5.2.1 二维数组的定义	138
5.2.2 二维数组元素赋值	138
5.3 字符数组的定义和应用	143
5.3.1 字符数组的定义及初始化	143
5.3.2 字符数组的输入和输出	145
5.3.3 字符串处理函数	149
5.4 编程实战	155
习题五	161
第六章 函数	168
6.1 函数的分类和定义	169
6.1.1 函数的分类	169
6.1.2 函数的定义	170
6.2 函数的返回值和参数	172
6.2.1 函数的返回值	172
6.2.2 函数的参数	174
6.2.3 参数的传递方式	180
6.3 嵌套调用	183
6.3.1 一般的嵌套调用	184
6.3.2 递归调用	187
6.4 变量的作用域	190
6.4.1 变量的作用域	190
6.4.2 变量的存储类别	198
6.5 函数的作用域	202
6.5.1 外部函数	202
6.5.2 内部函数	203
6.6 编程实战	203
习题六	208
第七章 指针	217
7.1 变量与指针	217
7.1.1 地址与指针	217
7.1.2 指向变量的指针变量	218
7.1.3 指针变量作函数参数	223
7.2 数组与指针	226
7.2.1 指针与一维数组的联系	226
7.2.2 指针与二维数组的联系	230
7.2.3 指向数组的指针作函数参数	234
7.3 字符串与指针	238

7.3.1 字符指针变量的定义与初始化	238
7.3.2 字符指针作函数参数	242
7.4 函数与指针	245
7.4.1 指向函数的指针	245
7.4.2 返回指针的函数	246
7.5 指针数组和多级指针	248
7.5.1 指针数组	248
7.5.2 多级指针	250
7.5.3 main 函数的形参	252
7.6 编程实战	253
7.7 指针小结	258
习题七	258
第八章 结构体与共用体	267
8.1 结构体概述	267
8.1.1 结构体类型的定义	267
8.1.2 结构体变量的定义	269
8.1.3 结构体变量的引用	269
8.1.4 结构体变量的赋值和初始化	270
8.2 结构体数组	270
8.3 指向结构体的指针	272
8.3.1 指向结构体的指针变量的定义	272
8.3.2 利用结构体指针变量引用成员	273
8.3.3 用结构体类型作函数参数	274
8.4 用结构体处理链表	275
8.4.1 链表概述	275
8.4.2 链表的建立与输出	276
8.4.3 链表的插入	278
8.4.4 链表的删除	279
8.4.5 链表的综合操作	280
8.5 共用体	283
8.5.1 共用体类型及变量的定义	283
8.5.2 共用体变量的引用	284
8.6 枚举类型	286
8.7 用 typedef 定义类型别名	287
习题八	288
第九章 预处理命令、位运算与文件	290
9.1 预处理命令	290

9.1.1 宏定义	290
9.1.2 文件包含	294
9.2 位运算	295
9.2.1 位运算符	295
9.2.2 位域	299
9.3 文件	302
9.3.1 文件和文件指针	302
9.3.2 文件的打开与关闭	303
9.3.3 文件的读写	305
9.3.4 文件的定位	309
9.3.5 文件的检测	311
习题九	311
综合训练	317
附录一 运算符及其优先级顺序表	326
附录二 标准 ASCII 码表	328
附录三 C 语言常用库函数一览表	329
习题答案	336
参考文献	340

第一章 引言

1.1 学习编程有什么用

学习编程有什么用?大多数非计算机专业的学生在学习程序设计类课程时都会不由自主地问这个问题。有的同学认为自己学的是化工专业,平时做得最多的就是化学实验,即使要用到一两个软件,也都是现成的,根本不需要亲自动手编程,那学习编程还有什么用呢?

其实不然,对于非计算机专业的学生而言,学习编程的目的并不是为了掌握某一门程序设计语言,也不是为了今后搞程序开发,而是为了锻炼逻辑推理和计算思维能力,通过了解计算机工作原理,提高用计算机解决问题的能力。

当今社会,计算机正在以各种各样的形式出现在我们周围。不管在学习中,还是在工作中,计算机都成为人们不可缺少的工具。计算机之所以具有这么多用途,正是因为有各种各样的软件存在。软件是计算机的灵魂,程序是软件的核心。在现实生活中,程序无处不在。当我们用 WPS、Office 写论文、处理数据或制作幻灯片时,用 QQ、微信和朋友聊天时,在网上淘宝时……都离不开程序。面对日益发达的信息社会,作为当代大学生,了解编程的奥秘是非常有必要的。

实际上,编程是一件非常美妙的事情。编程的乐趣是其他学科无法比拟的。一旦你亲身体会过程序运行成功所带来的那种喜悦,那种成就感,就会自然而然地迷恋上编程,就会觉得自己非常棒,居然能指挥计算机做事了。那种兴奋感将大大超越去玩别人预先设定好的游戏软件所能获得的。程序可以让你突然迸发的灵感和想法瞬间得以实现,并看到立竿见影的效果。这种即时性的反馈会让学习兴趣变得越来越浓厚。比尔·盖茨、乔布斯等 IT 巨人们在他们还只是少年的时候就已经疯狂地迷恋上编程了。正是编程,实现了他们的奇思妙想,给世界带来了一个又一个奇迹。

总之,学点编程绝对没有坏处,正所谓“艺多不压身,艺高人胆大”。在各行各业都要用到计算机的情况下,计算机编程也算是一技之长,对于提高就业竞争力大有帮助。

1.2 C 语言的发展历程

C 语言是 1972 年由美国的 Dennis M. Ritchie(丹尼斯·里奇,C 语言之父,UNIX 之父)设计发明的。1978 年美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时 Brian. W. Kernighan 和 Dennis M. Ritchie 合著了著名的《The C Programming Language》(《C 程序设计语言》)一书。现在此书已被翻译成多种语言,成为 C 语言方面最权威的教材之一。与此同时,C 语言也衍生了很多不同版本。1983 年美国国家标准局(American National Standards Institute,简称 ANSI)成立了一个委员会来制定 C 语言标准。1989 年 C 语言标准被批准,被称为 ANSI X3.159—1989 “Programming Language C”。这个版本的 C 语言标准通常被称为 ANSI C。

C 语言功能强大,既有高级语言的特点,又有汇编语言的特点,既可以作为系统设计语言编写系统程序,又可以作为应用程序设计语言编写应用程序。早期的 C 语言主要是用于 UNIX 操作系统。自 20 世纪 80 年代开始,C 语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的应用,成为当代最优秀的程序设计语言之一。当今社会随着个人电子消费产品和开源软件的流行,C 语言由于在底层控制和性能方面的优势,成为芯片级开发(嵌入式)和 Linux 平台开发的首选语言。在通信、网络协议、破解、3D 引擎、操作系统、驱动、单片机、手机、PDA(Personal Digital Assistant,个人数字助理)、多媒体处理、实时控制等领域,C 语言正在用一行行代码证明它从应用级开发到系统级开发的强大和高效。

1.3 C 语言的特点

1. 语言简洁紧凑、灵活方便

C 语言一共只有 32 个关键字和 9 种控制语句,只需用规范的方法就可以构造出功能很强的数据类型、语句和程序结构。程序简洁,表示方法简单,尽可能地压缩了一切不必要的成分,如用 $i++$ 或 $i+=1$ 表示 $i=i+1$, $--i$ 表示 $i=i-1$ 等。

2. 数据结构丰富

C 语言包含了多种数据结构,如整型、实型、字符型、数组、指针类型、结构体类型和共用体类型等,可以用于实现各种复杂的数据类型运算。

3. 运算符丰富

C 语言的运算符有 34 种,涉及的范围很广,灵活使用各种运算符可以实现难度极大的运算。

4. 语法检查不太严格

一般的高级语言对语法检查非常严格,几乎可以检查出所有的语法错误,但是 C 语言的语法检查不太严格,因此程序书写的形式可以比较自由。然而正因为编译器不能检查出所有的错误,所以编程人员要自己认真检查程序,以保证程序正确运行。

5. 结构化程序设计

C 语言是一种结构化语言,符合现代编程风格的要求。它用函数作为程序设计的基本单位,以实现程序的模块化。同时,它还提供了编写结构化程序所需的控制流语句,如 `if...else`、`for`、`while`、`do...while` 等。

6. 具有低级语言功能

C 语言允许直接访问物理地址,能够对二进制位、字节和地址进行操作,可以直接对硬件进行操作,具有低级语言的许多功能,因此既可用于开发应用软件,又可用于编写系统软件和单片机程序。

7. 生成的目标代码质量高

C 语言编写的程序比汇编语言编写的程序的可读性好,易于调试和修改,生成的目标代码质量与汇编语言生成的相当,程序执行效率只比汇编语言程序低 10% ~ 20%。

8. 可移植性好

与汇编语言相比,用 C 语言开发单片机系统软件具有很多优点,如软件调试直观、维护升级方便、代码的重复利用率高、便于跨平台的代码移植等。

1.4 如何学习 C 语言

学习一门新的程序设计语言的最佳途径就是阅读实例程序、模仿实例程序编写程序,通过各种实例程序来学习 C 语言的语法规则、编程的思路和技巧。对于教材中提到的所有实例程序,都要仔细研读,直到对每一行程序代码都理解了,然后合上书本,在计算机上把这些程序代码输进去并运行,再找几个类似的编程题目,尝试自己写出这段程序来,如果写不出来,回过头来继续研究实例程序,想想自己为什么写不出来,然后再去写这段程序,反反复复,直到你对于这些编程题目手到擒来为止。学习 C 语言贵在坚持上机实践,不能光看不练,在看书的同时就要敲代码,程序实际运行的各种情况可以让你更快更牢固地掌握知识点。只有通过反复的观察、分析、比较、实践、总结,才能熟练掌握 C 语言的各种数据类型、运算符、语句结构的使用方法,才能逐渐地积累编程经验,明白编程的奥秘,才能把书本上的知识变成自己的。

具体而言,学习编程要掌握以下步骤和要点:

- (1) 阅读教材上的案例,并按照教材上的案例编写程序,上机运行,查看运行结果。
- (2) 写几个小程序解决一些简单问题,以熟悉基本的算法和语法知识。
- (3) 从自己身边熟悉的事件入手,如成绩统计,可以先编制一小段程序,完成一个小功能,然后逐步完善。
- (4) 反复上机练习。C 程序设计是一门实践性很强的课程,只有多上机练习才能提高编程能力,而不能只满足于课堂听讲。还要注意课后作业是否会做,上机编程是否能得到正确结果,程序出现问题后能否调试出正确结果等。只有亲自动手编程才能真正提升自己的计算思维能力,增强自己用计算机解决问题的实践能力,为今后工作奠定坚实的基础。
- (5) 循序渐进。高级语言的语句看起来很多,也很复杂,对编程技巧要求也很高,但是其基本的语句和常用算法并不多,所以在学习过程中要注意对一些常用的基本算法进行归纳和总结,在熟练掌握基本语句和经典算法的基础上再进行举一反三、逐步深入。
- (6) 粗细结合。高级语言语法规则繁多,需要记忆的内容也很多,比如整数的范围、实数的范围,它们在机器中存储时所占的字节数,输出时的有效位数、默认宽度等,但是这些细节并不影响高级语言的初步学习,有些规则可通过上机练习逐步掌握,没必要死记硬背,可以粗一些。然而对于基本的函数和语言的书写格式、作用、执行过程等一定要记准并会灵活应用,这里就一定要细一些,因为高级语言的语法是非常严格的,一点小错误,如漏掉一个分号等,都会造成程序错误而无法运行。
- (7) 多种手段交流。通过 C 语言编程论坛、教学网站或 QQ 群与老师、同学、网友等进行学习交流,可以不断吸取别人的长处,丰富编程经验,提高编程水平。

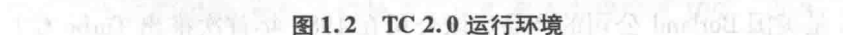
总之,学好 C 语言贵在实践出真知,掌握这一原则就一定会学好 C 语言的。

1.5 Turbo C 2.0 简介

Turbo C 是美国 Borland 公司的产品。该公司在 1987 年首次推出 Turbo C 1.0 产品,其中使用了集成开发环境,将文本编辑、程序编译、连接以及程序运行一体化,大大方便了程序的开发。1989 年 Borland 公司又推出 Turbo C 2.0(以下简称 TC 2.0 或 TC),在原来集成开发环境的基础上增加了查错等功能。TC 2.0 是一个使用广泛、高效快捷的 C 编译系统。它

1. 运行 TC 软件

TC 2.0 无需安装



十女排到女工腰样工之

下一行为功能键参考行,这四个部分构成了 TC 2.0 的主界面。按 F6 功能键可以在 Edit 窗口和 Message 窗口间切换,按 F5 功能键可以关闭或打开 Message 窗口。

Edit 窗口的顶端为状态行,其中:

- (1) Line 1 Col 1:显示光标所在的行号和列号,即光标位置。
- (2) Insert:表示编辑状态处于“插入”,当处于“改写”状态时此处为空白。按键盘上的 Insert 键可以在两种状态中进行切换。
- (3) Indent:自动缩进开关。在开启状态下,每一次换行将使光标自动与上一行的第一个字符对齐,可用 Ctrl + QI 组合键切换。
- (4) Tab:制表开关,可用 Ctrl + OT 组合键切换。
- (5) Fill:它与 Indent 和 Tab 的开关(ON/OFF)一起使用。当 Tab 模式为 ON 时,编辑系统将在每一行的开始填以适当的制表符及空格符。
- (6) Unindent:在该状态下,当光标处于某行的第一个非空字符或一空行时,退格键将使光标回退一级而不是一个字符,可用 Ctrl + OU 组合键切换。
- (7) NONAME.C:显示当前正在编辑文件的名称,显示为“NONAME.C”时,表示用户尚未给文件命名。

2. 配置运行环境

第一次使用 TC 软件时需要对运行环境进行配置。配置方法如下:按下 Alt + O 快捷键,打开 Options 菜单,选择 Options | Directories 菜单命令,将 Turbo C directory、Include directories、Library directories 分别指定为 TC、INCLUDE(头文件)、LIB(库文件)所在目录。在图 1.1 中,TC 软件文件所在目录为 C:\TC,头文件所在目录为 C:\TC\INCLUDE,库文件所在目录为 C:\TC\LIB,其环境配置如图 1.3 所示。



图 1.3 TC2.0 运行环境配置

环境配置好以后,选择 Options | Save options 菜单命令,将当前所做的配置保存在“TC-CONFIG.TC”文件中,这样以后打开 TC 2.0 时就不需要再进行配置了。

3. 编辑程序

启动 TC 之后光标会停留在 File 菜单上,此时可按下 Esc 键或 F10 功能键来激活 Edit 窗口。按下 Alt + F 快捷键打开 File 菜单,选择其中的 New 子菜单项可以编辑新的源程序,如

图1.4和1.5所示。

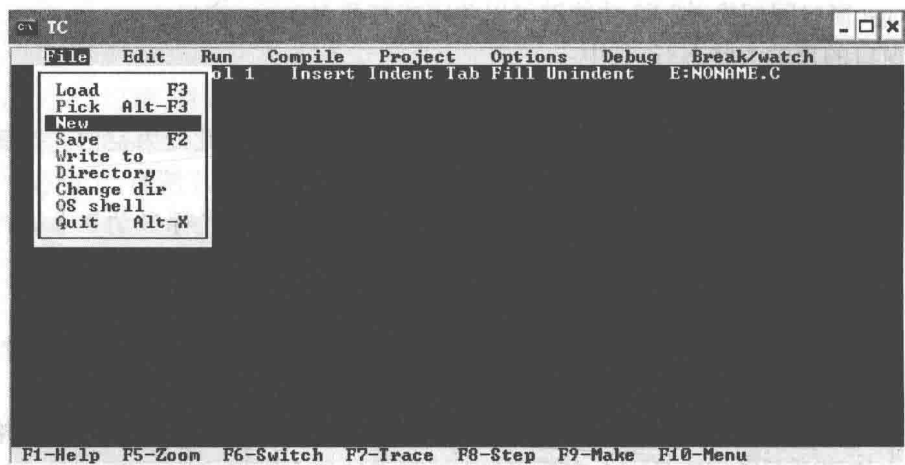


图 1.4 创建新文件

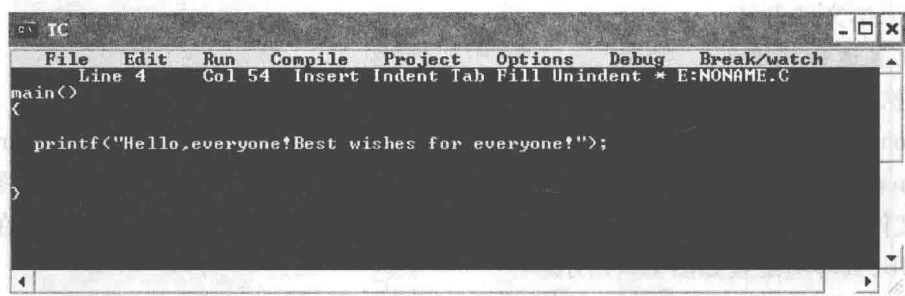


图 1.5 编辑源程序

4. 编译程序

选择 Compile|Compile to OBJ 菜单命令对源程序进行编译,编译成功后再选择 Compile|Link EXE file 菜单命令生成 EXE 文件,如图 1.6 和图 1.7 所示。



图 1.6 编译源程序



图 1.7 连接库函数并生成 EXE 文件

编译和连接这两个步骤可以合成一步完成,即选择 Compile | Make EXE file 菜单命令或直接按 F9 功能键都可以完成编译和连接操作。

程序经编译后如果有错会在 Message 窗口中显示错误信息,按 F6 功能键可以返回 Edit 窗口修改程序。

5. 运行程序

按 Alt + R 快捷键选择 Run 菜单或直接按 Ctrl + F9 快捷键都可以运行程序,如图 1.8 所示。



图 1.8 运行程序

6. 查看结果

程序运行完后选择 RUN 菜单下的 User screen 命令或按 Alt + F5 快捷键可以看到程序运行结果,如图 1.9 所示。

如果在程序末尾加一句“getch();”,则在程序运行之后可以直接看到结果。

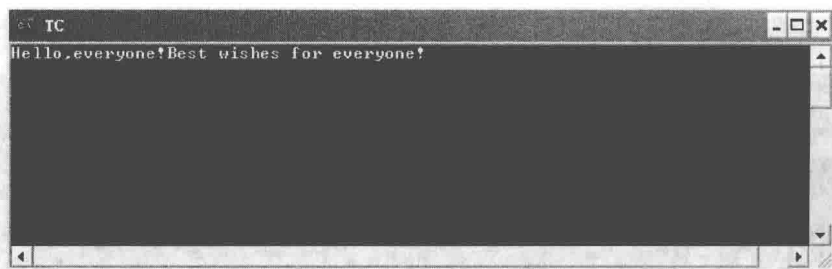


图 1.9 程序运行结果

7. 保存文件

选择 File | Save 菜单命令或直接按 F2 快捷键都可以保存源程序文件,保存文件时文件的扩展名必须为“.c”,如图 1.10 所示。

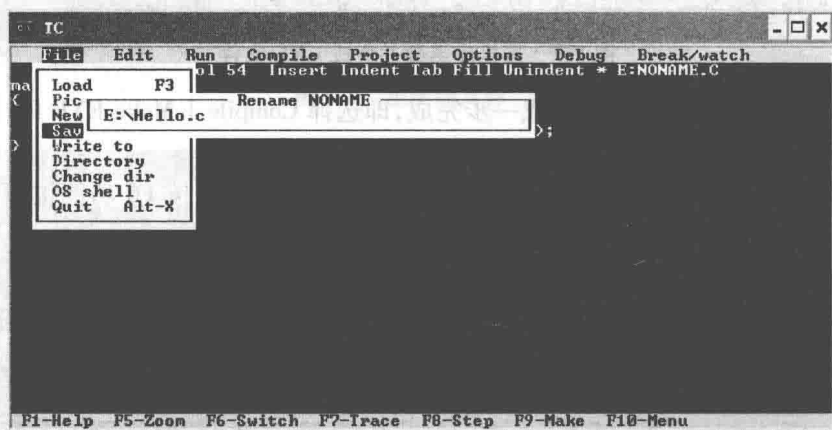


图 1.10 保存源程序文件

8. 打开文件

选择 File | Load 菜单命令或直接按 F3 快捷键可以打开一个已经存在的源程序文件,如图 1.11 所示。



图 1.11 用 Load 命令打开源程序文件