



教育部职业教育与成人教育司推荐教材  
中等职业学校计算机技术专业教学用书

# C语言案例教程

## (第2版)

© 白炽贵 编著

本书配有电子教学参考资料包



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

教育部职业教育与成人教育司推荐教材  
中等职业学校计算机技术专业教学用书

# C 语言案例教程

## (第2版)

白炽贵 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书为中等职业学校C语言程序设计课程的试用教材,是结合中等职业学校的教学实际编写的,适合于中等职业教育计算机应用专业教学使用。

本书按循序渐进、逐步到位的笔法写成。全书共分11章。第1章用“移动的金字塔”这一具有三重循环的简易动画程序,来进行编程拉练,以鸟瞰C语言编程的概貌。第2章至第8章主要介绍C语言基础,运算符和表达式、顺序结构、选择结构、循环结构、数组和函数。第9章为编程实训,该实训用7个由小变大的例题程序和4个空函数的功能完善组成11级阶梯,通过在阶梯每一步上的算法分析和上机操作,使学生实训从只能输出某一固定方块的简单程序,一步一步攀登上了整个游戏功能全部实现的复杂程序。第10章简要介绍了结构体和指针及共用体。第11章介绍了著名的汉诺塔问题和相应的游戏程序设计。

本书配有学习光盘,光盘中主要提供了43个教学课件和书中所有例题程序及习题程序的源代码,对自学极有帮助,还可作为广大中学生和C语言初学者的自学教材。

本书配有电子教学资料包,包括教学指南、电子教案、习题答案、详见前言。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

C语言案例教程/白炽贵编著. —2版. —北京:电子工业出版社,2009.6  
教育部职业教育与成人教育司推荐教材·中等职业学校计算机技术专业教学用书  
ISBN 978-7-121-08550-5

I. C… II. 白… III. C语言—程序设计—专业学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2009)第041654号

责任编辑:肖博爱 特约编辑:王新永

印 刷:涿州市京南印刷厂

装 订:涿州市桃园装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本:787×1092 1/16 印张:14.5 字数:371.2千字

印 次:2009年6月第1次印刷

印 数:3000册 定价:25.70元(含光盘1张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。



本书为 C 语言程序设计课的试用教材,是结合中等职业教育的教学实际编写的,适合各中职学校计算机应用专业教学使用。

趣味 C 语言、人文 C 语言、实训 C 语言,是本书的编著目标。尊重学生情感,贴近学生生活,是本书以学生为本的守则。本书针对中等职业学校计算机应用专业教学实际,在 C 语言的语法规则说明上追求精练,在编程实训上注重力度。本书在考虑趣味性、操作性的同时,把基础知识的编程应用做了精心设计和强化提高。本书共分为 11 章。前 9 章为必学内容,第 10 章为选学内容,第 11 章可安排为自学内容。

第 1 章用“移动的金字塔”动画程序来进行编程拉练,以此鸟瞰 C 语言编程的概貌和感悟 C 语言的学习方法。

第 2 章至第 8 章为编程基础,主要介绍了 C 语言基础、运算符和表达式、顺序结构、选择结构、循环结构、数组和函数等基本知识。

第 9 章把俄罗斯方块这个经典游戏引入了编程实训的内容。面对这个有很大难度的程序设计,实训采用了循序渐进、一步一个台阶的讲解笔法:用 7 个由小变大的例题程序和 4 个空函数的功能完善组成 11 级阶梯,对俄罗斯方块实训编程的思路和步骤做了具体详尽的分析,使读者亲身经历了个从基础知识的基本学习到基础知识的深入应用的学习升华过程。

第 10 章简要介绍了结构体和指针及共用体这三个重要概念。

第 11 章介绍了著名的汉诺塔问题和相应的游戏程序设计。

本书所有源程序以 Turbo C2.0 为编译工具;借助 UC DOS98 在程序中使用汉字。操作系统可用 Windows XP、Windows 98 及 DOS6.22 等。

为了方便教学,本书配有学习光盘。光盘中主要提供了 43 个教学课件。这些课件以动画的形式演示了用语言或文字难以叙述清楚的程序具体运作过程,具有很强的助教助学功能。受教学课时的限制,本书没有讲解顺序查找、折半查找、希尔排序、插入排序、快速排序等经典内容,而是用光盘课件的形式进行介绍。课件形式更为直观形象,适用于自学。另外,光盘中还提供了书中所有例题程序和习题程序的源代码,可复制到相应目录中进行学习和调试,以节省输入源程序的时间。

本书在编写过程中,作者参考了谭浩强教授的“C 程序设计”,借鉴了杨锦良先生关于方块

符号、网上同行关于  $4 \times 4$  网格的使用经验,得到了税清贵高级教师的热情帮助和指教。初稿完成后,袁泽中教研员认真审读全书并提出了极为重要的修改意见。另外,本书从下笔伊始至书稿完成文印及教学试用,一直得到学校教育处的大力支持和鼓励。在此,谨向他们表示感谢。同时,由于编著者水平有限,书中不妥之处在所难免,敬请读者批评指正。

本书配有电子教学资料包,包括教学指南、电子教案、习题答案。请有此需要的教师登录华信教育网([www.hxedu.com.cn](http://www.hxedu.com.cn))下载,或与电子工业出版社联系(E-mail:ve@phei.com.cn),我们将免费提供。

白炽贵

2009年5月



# 目 录



|                                     |    |
|-------------------------------------|----|
| 第 1 章 编程拉练：移动的金字塔 .....             | 1  |
| 1.1 从画一个“*”号起步 .....                | 2  |
| 说明 1: UC DOS+TC2.0 运行环境的建立和启动 ..... | 2  |
| 说明 2: C 程序的上机运行步骤 .....             | 4  |
| 1.2 画点成线 .....                      | 7  |
| 1.3 画线成形 .....                      | 8  |
| 1.4 学会定位光标 .....                    | 9  |
| 1.5 画出金字塔 .....                     | 10 |
| 1.6 让金字塔向右匀速运动 .....                | 11 |
| 1.7 让金字塔沿方圈移动 .....                 | 13 |
| 1.8 探究程序的执行流程 .....                 | 15 |
| 小结 1 .....                          | 16 |
| 习题 1 .....                          | 17 |
| 第 2 章 C 语言基础 .....                  | 18 |
| 2.1 C 语言的元字符和标识符 .....              | 18 |
| 2.2 C 语言的关键字和语句 .....               | 18 |
| 2.3 C 源程序结构 .....                   | 19 |
| 2.4 C 语言的数据类型 .....                 | 20 |
| 2.5 常量和变量 .....                     | 20 |
| 2.6 整型数据 .....                      | 21 |
| 2.6.1 整型常量 .....                    | 21 |
| 2.6.2 整型变量 .....                    | 22 |
| 2.7 实型数据 .....                      | 23 |
| 2.7.1 实型常量 .....                    | 23 |
| 2.7.2 实型变量 .....                    | 23 |
| 2.8 字符型数据 .....                     | 24 |
| 2.8.1 字符常量 .....                    | 24 |
| 2.8.2 字符变量 .....                    | 26 |
| 2.8.3 字符串常量 .....                   | 27 |
| 2.9 数据类型转换 .....                    | 27 |
| 2.9.1 自动类型转换 .....                  | 27 |

|                                 |    |
|---------------------------------|----|
| 2.9.2 强制类型转换 .....              | 28 |
| 2.10 数据的输出与输入 .....             | 28 |
| 2.10.1 格式输出函数 printf() .....    | 28 |
| 2.10.2 格式输入函数 scanf() .....     | 30 |
| 2.10.3 字符输出函数 putchar() .....   | 31 |
| 2.10.4 字符输入函数 getchar() .....   | 31 |
| 2.10.5 无回显的字符输入函数 getch() ..... | 32 |
| 小结 2 .....                      | 33 |
| 习题 2 .....                      | 33 |
| <b>第 3 章 运算符与表达式</b> .....      | 35 |
| 3.1 算术运算符与算术表达式 .....           | 35 |
| 3.2 赋值运算符与赋值表达式 .....           | 37 |
| 3.3 关系运算符与关系表达式 .....           | 40 |
| 3.3.1 关系运算符 .....               | 40 |
| 3.3.2 关系表达式 .....               | 40 |
| 3.4 逻辑运算符与逻辑表达式 .....           | 41 |
| 3.5 逗号运算符与逗号表达式 .....           | 46 |
| 3.6 运算符的优先级和结合性 .....           | 47 |
| 小结 3 .....                      | 47 |
| 习题 3 .....                      | 48 |
| <b>第 4 章 顺序结构程序设计</b> .....     | 50 |
| 4.1 C 语句的分类 .....               | 50 |
| 4.2 算法的概念 .....                 | 51 |
| 4.3 算法的流程图表示 .....              | 51 |
| 4.4 程序的 3 种基本结构 .....           | 52 |
| 4.5 顺序结构程序设计示例 .....            | 53 |
| 小结 4 .....                      | 55 |
| 习题 4 .....                      | 55 |
| <b>第 5 章 选择结构</b> .....         | 58 |
| 5.1 if 语句 .....                 | 58 |
| 5.2 if...else 语句 .....          | 60 |
| 5.3 if...else if 语句 .....       | 61 |
| 5.4 switch 语句 .....             | 63 |
| 5.5 选择结构的嵌套 .....               | 66 |
| 5.5.1 if 语句的嵌套 .....            | 66 |
| 5.5.2 switch 语句的嵌套 .....        | 66 |
| 5.5.3 if 嵌套的应用 .....            | 67 |
| 小结 5 .....                      | 70 |
| 习题 5 .....                      | 70 |

|                                 |     |
|---------------------------------|-----|
| 第 6 章 循环结构 .....                | 72  |
| 6.1 for 语句 .....                | 72  |
| 6.1.1 for 语句的一般形式 .....         | 72  |
| 6.1.2 for 语句的变式 .....           | 77  |
| 6.2 while 语句 .....              | 79  |
| 6.3 do...while 语句 .....         | 81  |
| 6.4 循环结构的三个专用术语 .....           | 84  |
| 6.5 循环结构应用实例: 飞雪迎春 .....        | 86  |
| 6.6 循环的嵌套 .....                 | 87  |
| 6.7 break 语句和 continue 语句 ..... | 89  |
| 6.7.1 break 语句 .....            | 89  |
| 6.7.2 continue 语句 .....         | 91  |
| 小结 6 .....                      | 92  |
| 习题 6 .....                      | 92  |
| 第 7 章 数组 .....                  | 95  |
| 7.1 从排序谈数组 .....                | 95  |
| 7.2 一维数组的定义和引用 .....            | 96  |
| 7.2.1 一维数组的定义 .....             | 96  |
| 7.2.2 一维数组元素的引用 .....           | 97  |
| 7.2.3 一维数组的初始化 .....            | 98  |
| 7.2.4 数组元素的赋初值 .....            | 98  |
| 7.3 一维数组应用 .....                | 98  |
| 7.3.1 数据统计 .....                | 98  |
| 7.3.2 数据排序 .....                | 99  |
| 7.4 二维数组 .....                  | 103 |
| 7.5 二维数组应用示例 .....              | 104 |
| 7.6 一维字符数组 .....                | 105 |
| 7.6.1 一维字符数组的定义 .....           | 106 |
| 7.6.2 字符串常量和字符串结束标志 .....       | 106 |
| 7.6.3 一维字符数组的初始化 .....          | 106 |
| 7.6.4 字符数组的应用 .....             | 106 |
| 7.7 二维字符数组 .....                | 109 |
| 7.8 “嫦娥探月” .....                | 111 |
| 小结 7 .....                      | 115 |
| 习题 7 .....                      | 116 |
| 第 8 章 函数 .....                  | 120 |
| 8.1 函数的基本概念 .....               | 120 |
| 8.1.1 系统库函数 .....               | 120 |
| 8.1.2 用户自定义函数 .....             | 120 |

|               |                              |     |
|---------------|------------------------------|-----|
| 8.2           | 函数的定义 .....                  | 121 |
| 8.2.1         | 无参数函数的定义 .....               | 121 |
| 8.2.2         | 有参数函数的定义 .....               | 121 |
| 8.2.3         | 空函数 .....                    | 122 |
| 8.3           | 形式参数和函数的值 .....              | 122 |
| 8.4           | 函数的调用 .....                  | 123 |
| 8.4.1         | 函数调用的一般形式 .....              | 123 |
| 8.4.2         | 函数调用的方式 .....                | 123 |
| 8.4.3         | 对被调函数的声明 .....               | 124 |
| 8.5           | 函数调用的参数传递 .....              | 125 |
| 8.6           | 局部变量和全局变量 .....              | 128 |
| 8.6.1         | 局部变量 .....                   | 128 |
| 8.6.2         | 全局变量 .....                   | 129 |
| 8.7           | 函数的嵌套调用和递归调用 .....           | 131 |
| 8.8           | “抗震救灾众志成城”主题动画显示 .....       | 133 |
| 小结 8          | .....                        | 137 |
| 习题 8          | .....                        | 137 |
| <b>第 9 章</b>  | <b>编程实训：俄罗斯方块</b> .....      | 141 |
| 9.1           | 俄罗斯方块游戏中各方块的坐标表示 .....       | 141 |
| 9.2           | 方块坐标的变量存取方式 .....            | 144 |
| 9.3           | 方块坐标的数组存取方式 .....            | 145 |
| 9.4           | 把画块操作设计为专用函数 .....           | 146 |
| 9.5           | 增强画块函数的适应性 .....             | 148 |
| 9.6           | 在约定位置随机产生新方块 .....           | 149 |
| 9.7           | 方块的预设移动 .....                | 151 |
| 9.8           | 方块的键控移动 .....                | 153 |
| 9.9           | 俄罗斯方块游戏的整体结构设计 .....         | 156 |
| 9.10          | 俄罗斯方块游戏的整体框架程序 .....         | 157 |
| 9.11          | 完善框架程序中的可移性检测功能 .....        | 164 |
| 9.12          | 完善框架程序中的初始化功能 .....          | 165 |
| 9.13          | 完善框架程序中的方块落底处理功能 .....       | 168 |
| 9.14          | 完善框架程序中的更新显示功能 .....         | 170 |
| 9.15          | 场景数组与游戏区域的同屏显示 .....         | 172 |
| 9.16          | 完整的俄罗斯方块游戏程序 .....           | 172 |
| <b>第 10 章</b> | <b>C 语言的高级特性</b> .....       | 179 |
| 10.1          | 结构体 .....                    | 179 |
| 10.1.1        | 结构体类型定义和结构体变量、结构体数组的定义 ..... | 179 |
| 10.1.2        | 结构体变量和结构体数组的使用 .....         | 181 |
| 10.2          | 指针 .....                     | 183 |

|               |                                |     |
|---------------|--------------------------------|-----|
| 10.2.1        | 变量的地址和指针概述 .....               | 183 |
| 10.2.2        | 指针变量的定义和引用 .....               | 183 |
| 10.2.3        | 指向字符串的指针 .....                 | 185 |
| 10.2.4        | 指向结构体变量的指针 .....               | 187 |
| 10.2.5        | 指向结构体数组的指针 .....               | 188 |
| 10.2.6        | 指向函数的指针 .....                  | 190 |
| 10.2.7        | 关于指针运算的一个说明 .....              | 191 |
| 10.3          | 结构体和指针的应用: 渡河问题 .....          | 193 |
| 10.4          | 共用体类型简介 .....                  | 201 |
| 小结 10         | .....                          | 202 |
| 习题 10         | .....                          | 203 |
| <b>第 11 章</b> | <b>编程阅读: 演玩汉诺塔</b> .....       | 204 |
| 11.1          | 汉诺塔问题 .....                    | 204 |
| 11.2          | 演玩汉诺塔 .....                    | 207 |
| 11.2.1        | 游戏界面的布局设计 .....                | 207 |
| 11.2.2        | 游戏进程的思路设计 .....                | 208 |
| 11.2.3        | 游戏功能的模块设计 .....                | 208 |
| 11.2.4        | 源程序 .....                      | 208 |
| 11.2.5        | 各模块功能设计要点简介 .....              | 216 |
| <b>附录 A</b>   | <b>C 语言中的关键字</b> .....         | 220 |
| <b>附录 B</b>   | <b>文本屏幕颜色值的定义</b> .....        | 221 |
| <b>附录 C</b>   | <b>常用字符与 ASCII 代码对照表</b> ..... | 222 |

# 第1章 编程拉练：移动的金字塔



第1章的学习任务，就是完成“移动的金字塔”程序设计。即把下面这个用字符“\*”来构成的一个简单图形沿一个矩形路线运动一圈。这个程序具有动画运行结果和层次分明的程序结构。它充分展现出了流程控制这一程序设计的核心思想，最适合作为初学者的入门例题。完成这一项目的程序设计实践后，能够基本了解C语言编程的概要过程和粗略方法，从而为后面各章的深入学习触发出一些有用的启迪。

```
      *
     ***
    *****
   *********
```

怎样进行这一程序设计呢？先从对此问题的一般分析说起。要使图形运动，首先是要画有图形，也就是首先要用程序来画出那个图形。怎样来画出上面那个图形呢？可以看到，那个金字塔基本上是由线组成的，而线又是由字符“\*”组成的，因此就要首先从画“\*”开始，学习如何用C语言来进行程序设计。

完成这一程序设计只需很少的C语言知识。在本章中，只针对能完成这个任务的很少一部分基本内容，进行学习和实践。当成功地完成这个项目的编程任务后，就要用从中感悟出来的学习方法，从第2章起，开始学习C语言的基础知识，以形成完整的知识体系。由于这个程序是含有四个三重循环依次串接而成的组合形式，为降低难度，采用编程拉练的形式，来进行学习和实践。

编程拉练这种学习形式，就是把实现最终效果的程序，分解成由易到难，逐步扩充的几级台阶程序，从而一步一个台阶地进行程序设计。这次拉练编程的各台阶依次是

```
1.点      2.线      3.形      4.金字塔      5.金字塔右移      6.金字塔移一周
```

```
      *      ***      *****      *      *      *
      ***      *****      ***      ***      ***
     *****      *****      *****      *****      *****
    *****      *****      *****      *****      *****
   *****      *****      *****      *****      *****
  *****      *****      *****      *****      *****
 *****      *****      *****      *****      *****
```

具体该怎样用C语言来编写可画出上面那些图形的程序？千里之行，始于足下，让我们一起迈出这编程拉练的第一步。



## 1.1 从画一个 “\*” 号起步

【例 1.01】 在屏幕上画出一个 “\*” 号。

源程序:

```
#include<stdio.h>
main()
{
    putchar('*');
}
```

这是我们学习的第一个 C 语言程序。它的功能,就是在屏幕的当前光标位置上,输出一个点 “\*”。别看它只能画出一个点,正是从画点起步,逐步扩充,从而去画出那个移动的金字塔。因此,从一开始,就要认真对整个程序的每一行代码进行分析。

程序中的第一行称为编译预处理命令,预处理命令的特征是以 “#” 号打头,注意这行的结尾不能有分号。这个 `include` 预处理命令的作用是用来包含第 4 行那个 (函数调用) 操作所需的头文件,如果没有这一行,那对大括号中间的库函数调用在被编译时将会出现错误。这里,把它作为 C 语言的一个系统规则来遵守就可以了。

程序中的第二行到最后一行是这个程序中的主函数 `main()`。C 程序由主函数 `main` 和若干个 C 函数组成。C 函数就是子程序。C 语言规定: C 程序中必须有一个、且也只能有一个名为 `main` 的主函数; C 程序必须从 `main()` 函数开始运行, C 程序也必须在 `main()` 函数中结束运行。

第二行被称为主函数的函数头,其中 `main` 称为函数名,紧接其后的那对小括号是函数这种程序对象的标记, C 语言规定函数头上的小括号后不能有分号 “;”。

第三行到第五行是 `main()` 函数的函数体。第三行的大括号 “{” 是函数体的开始标志,第五行的大括号 “}” 是函数体的结束标志,它们必须成对使用。这对大括号之间的是组成函数体的 C 语句。语句是规定计算机完成某一步操作的指令。

第四行是调用 `putchar()` 库函数语句,它末尾的分号 “;” 是语句的结束标志。`putchar()` 函数的功能是,在屏幕上当前光标处,输出由小括号内参数所指定的一个 ASCII 字符。

任何源程序都必须经过系统编译并连接成可执行文件后才能运行而得到其结果,一个 C 语言程序需要通过上机编辑、编译、连接、运行来得到我们所预期的结果,因此先要建立运行环境。后面的内容将对如何上机运行程序等做两点补充说明。

### 说明 1: UC DOS+TC2.0 运行环境的建立和启动

尽管上面那个程序并未使用汉字,但为了在后面学习的程序中能输出汉字,因此考虑搭建汉字平台。在 Windows XP 等操作系统下使用 TURBOC2.0 (以下简称为 TC2.0) 时,应先运行 UC DOS,进入 UC DOS 后才进入 TC2.0,在硬盘上复制有 UC DOS 和 TC2.0 即可。现假定 UC DOS、TC2.0 均在 C 盘根目录下,且 TURBOC2.0 目录名已被改为 TC,则启动 UC DOS 和 TC2.0 的步骤如下。

在 Windows XP 桌面上,用鼠标依次单击 “开始” → “程序” → “附件” → “命令提示符”。当用鼠标单击的这四步完成后,就出现如图 1.1 所示的命令提示符界面图。

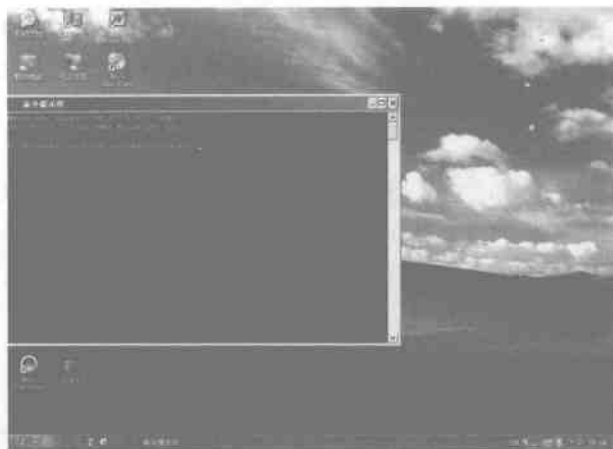


图 1.1

在命令提示符下，按如下顺序输入 DOS 命令（本书中用“<CR>”表示按一次回车键，并用加下画线表示从键盘输入）。

CD \UCDOS<CR>

RD16<CR>

KNL<CR>

PY<CR>

WB<CR>

CD\TC<CR>

TC<CR>

之后，便进入 Turbo C2.0 集成开发环境，屏幕上出现如图 1.2 所示的界面。



图 1.2

如图 1.2 所示，TC 集成开发环境的主屏幕由四部分组成：自上而下依次是主菜单、编辑窗口、信息窗口、功能键提示行。最下面一行不属于 TC 集成开发环境，它是 UC DOS 的输入法提示行；它接受右 Shift 键的开关，仅在源程序的编辑中要输入汉字时才开启，其他



情况下应予关闭。UCDOS 的输入法提示行开启时: 按“Alt+F2”快捷键切换为拼音输入; 按“Alt+F5”快捷键切换为五笔输入; 按“Alt+F6”快捷键切换为英文输入。对照图 1.2, 下面对 TC 集成开发环境主屏幕的四个组成部分做简单介绍。

#### 1) 主菜单行

主菜单行在主屏幕的顶行, 它共有 8 个菜单项, 分别是 File (文件)、Edit (编辑)、Run (运行)、Compile (编译)、Project (项目)、Options (选项)、Debug (调试)、Break/watch (断点/监视)。每个主菜单项还有下拉菜单, 分别用来实现各项操作。

#### 2) 编辑窗口

编辑窗口在主菜单下面, 占据了主屏幕的大部分面积, 该窗口用来输入和编辑 C 源程序。

#### 3) 信息窗口

信息窗口在编辑窗口的下面, 用来显示系统编译和连接时的有关信息。

#### 4) 功能键提示行

功能键提示行在信息窗口的下面, 显示有关功能键的作用。

在了解了 TC 集成环境的构成后, 现在来完成一个设置工作。首先按“F10”键激活主菜单, 然后将光标移到“Option”菜单上, 再选择其子菜单“Directories”进行目录参数设置, 即将原“C:\TURBOC2\INCLUDE”改为“C:\TC\INCLUDE”, 将原“C:\TURBOC2\LIB”改为“C:\TC\LIB”。完成修改后执行下面的子菜单“Save options”, 以保存各选项设置。

### 说明 2: C 程序的上机运行步骤

下面以例题 1.01 为例, 说明 C 语言源程序的有关上机步骤。

#### 1) 在 TC 集成开发环境中编辑源程序

进入 TC+UCDOS 环境后, 按 F10 键激活主菜单, 执行“File”→“New”, 即进入源程序的编辑状态, 此时的文件名暂为默认的 NONAME, 光标出现在编辑窗口的首字符位置, 这样就可开始按例 1.01 输入源程序代码。需要注意的是源程序的所有字符包括标点符号均应在纯英文半角状态下进行输入。编辑中若发现有误, 可操作光标键移动光标到需修改处, 进行字符的插入或修改, 如图 1.3 所示。



图 1.3



另外，也可用 File 菜单中的 Load 子菜单来打开一个已有的源程序。即在其弹出的对话框中输入文件名，如这个文件名已存在，则将其调入编辑窗口进行编辑；如果不存在，就创建一个新文件以对其编辑。这个子菜单的相应快捷键是[F3]。使用快捷方式要先关闭 UC DOS 的输入法提示行。

### 2) 在 TC 集成开发环境中将 C 源程序存盘

源程序输入编辑完成后，在运行前应先存盘，以防止运行失控而丢失文件。按 F10 键激活主菜单，执行菜单“File”→“Save”，在弹出的对话框中输入文件名如“LI01”（扩展名部分“.C”可以不输，TC 系统会自动添加），然后按回车键即可，如图 1.4 所示。



图 1.4

存盘操作的相应快捷键为[F2]，使用快捷方式要先关闭 UC DOS 提示行。

### 3) 在 TC 集成开发环境中运行 C 源程序和观察运行结果

用 F10 键激活主菜单，执行“Run”→“Run”，则系统就自动连续完成程序的编译、连接、运行，运行结束后自动返回到 TC 编辑屏幕。当编译或连接产生错误时，出错信息会显示在信息窗口中，用户可根据出错信息的提示对程序进行修改后再运行。由于程序成功运行后都要返回编辑屏幕，为了能看到程序运行时的输出结果，须将编辑屏幕切换为用户屏幕。方法是，用快捷键 F10 激活主菜单，执行“Run”→“User screen”，这样，就进入了如图 1.5 所示的用户屏幕，观察程序执行结果，按任意键之后又返回 TC 编辑屏幕。



图 1.5



以上两个操作的快捷键分别为“Ctrl+F9”和“Alt+F5”，且应在关闭 UC DOS 提示行的状态下时使用。

#### 4) 退出 TC 集成开发环境

按 F10 键激活主菜单，执行“File”→“Quit”即可退出 Turbo C 系统，相应的快捷键为“Alt+X”。

当我们按以上补充说明的方法和步骤上机，运行我们的第一个 C 程序而看到了一个“\*”号，如图 1.5 所示，说明我们已经胜利地迈出了编程拉练的第一步。

我们在看到这个“\*”号的同时，还看见了其他程序运行后残留在屏幕上的显示文本。要清除这些过时的显示内容，可在我们的程序中调用一个 C 库函数 clrscr()来解决，如下面例题中的程序。

**【例 1.02】** 在空白屏幕上画出一个“\*”。

源程序：

```
#include<stdio.h>
main()
{
    clrscr();          /* 4 */
    putchar('*');
}
```

这个程序与我们学习的第一个 C 程序相比，只多了第 4 行的代码“clrscr();”。在 C 语言程序中，形如“/\* \*/”格式的内容是注释，它对程序的编译和运行不起任何作用。

“/\*”和“\*/”必须成对使用，两者之间的文字只是给读程序的人看的。编译程序只要遇到这对符号，就对这对符号及其中的内容不做任何处理。在这里，只是用它来给我们说明这一行是第 4 行。前面的“/\*”和后面的“\*/”必须成对使用，且前面的“/”与“\*”、后面的“\*”与“/”都必须相连。否则为语法错误，程序就不能通过编译，也不可能运行。

在第 4 行上，确定计算机要进行某种具体操作的代码，是前面部分的“clrscr();”。此代码是一个调用清屏库函数 clrscr()的语句。clrscr()函数的功能是清屏，即在该调用语句执行时，将屏幕上已有的所有内容清除干净，从而使屏幕清晰地显示输出。

由于这个程序是在例 1.01 源程序 L101.c 基础上增加一个语句而成的，于是我们就可以先把源程序 L101.C 加载到 TC 的编辑窗口，再在其相应位置增加这一个“clrscr();”语句，并另存为“L102”，来得到第二个程序。这样做，不单是减少了键盘输入的工作量，而且还保证了新程序主体部分的正确性。因为这个新程序是在前一个完全正确的旧程序基础上扩建而成的，当这个新程序运行或调试中出现问题时，我们就可以把其问题的检查限定在所增添的代码范围内。这样可提高我们编程学习的效率。

把一个在当前编程环境即 TC 目录中已有的 C 源程序文件加载到编辑窗口中且另存为另一个 C 源程序文件的操作步骤如下：

激活主菜单，执行“File/Load”菜单，在弹出的对话框中，输入要载入的 C 源程序文件名，如“L101”，按回车键后再执行“File/Save to”菜单，又在弹出的对话框中输入新程序文件名，如“L102”，按回车键后，TC 编辑窗口中的源代码对应的文件名就改变为“L102”。这样就完成了在 TC 集成开发环境中，把一个 C 源程序文件另存为另一个 C 源程序文件。



此后，在编辑窗口的相应位置输入所要增添的代码，就比较简单地得到了第二个程序。在本章后面各个例题的上机操作中，都可以采用这种“另存为”的方法，由前一个源程序复制和修改而得到后一个程序，从而保证我们的编程学习总是在正确的基础上向前推进。

## 1.2 画点成线

把画点的语句重复执行，就由点构成线，下面的程序就是用这种方法来画一条线的。

**【例 1.03】** 重复地画出 7 个 “\*” 来形成一条线。

源程序：

```
#include<stdio.h>
main()
{
    int x;                /* 4 */
    clrscr();             /* 5 */
    for(x=1; x<=7; x++)   /* 6 */
        putchar('*');    /* 7 */
}
```

这个程序中的第 4 行是定义变量  $x$ 。变量是指在程序运行过程中其值可以改变的量。必须在这里先定义  $x$  后，下面的第 6 行才能够使用  $x$ ，这是 C 语言关于“变量必须先定义，后使用”的系统规则。

第 6、7 两行构成了一个基本的循环结构，其中第 6 行可称为循环头，第 7 行称为循环体。由循环头来控制循环体所需要进行重复操作的次数。

该循环头内有三个式子。第一个式子“ $x=1$ ”是给循环变量  $x$  赋初值 1；第二个式子“ $x<=7$ ”为循环条件，即其循环条件是  $x$  不大于 7；第三个式子“ $x++$ ”是循环变量增 1（就是由 1 变成 2、由 2 变成 3，依次类推）。在以下的程序解释中，为方便把这三个式子依次称为式子 1、式子 2、式子 3。

循环头对循环的控制规则：

程序执行流程是自上而下（即刚执行了第 5 行的“`clrscr();`”的操作后）进入循环头时，首先执行式子 1，即循环变量  $x$  被赋初值 1，紧接着就用这个初值为 1 的  $x$  去检查一次循环条件“ $x<=7$ ”，只要条件成立，就执行一次循环体，这就输出了一个点。

循环体执行完毕后，程序执行流程就自下而上返回循环头。此时在循环头中，是先执行式子 3，即进行循环变量  $x$  的增 1。增 1 后才执行式子 2，即重新检查循环条件“ $x<=7$ ”，条件成立就又执行循环体。循环体执行完毕就又自下而上返回循环头，又先执行式子 3（执行变量  $x$  的增 1），再执行式子 2（重新检查循环条件），……。当循环变量  $x$  由 7 增 1 成 8 后，循环条件“ $x<=7$ ”就不成立了，因此退出循环结构。

于是，在循环头的控制下，循环控制变量  $x$  从 1 变到了 8，变化过程中执行了 7 次循环体，变到 8 后因循环条件不再成立而退出整个循环结构，从而完成了连续画 7 个 “\*” 来形成一条线的控制任务。