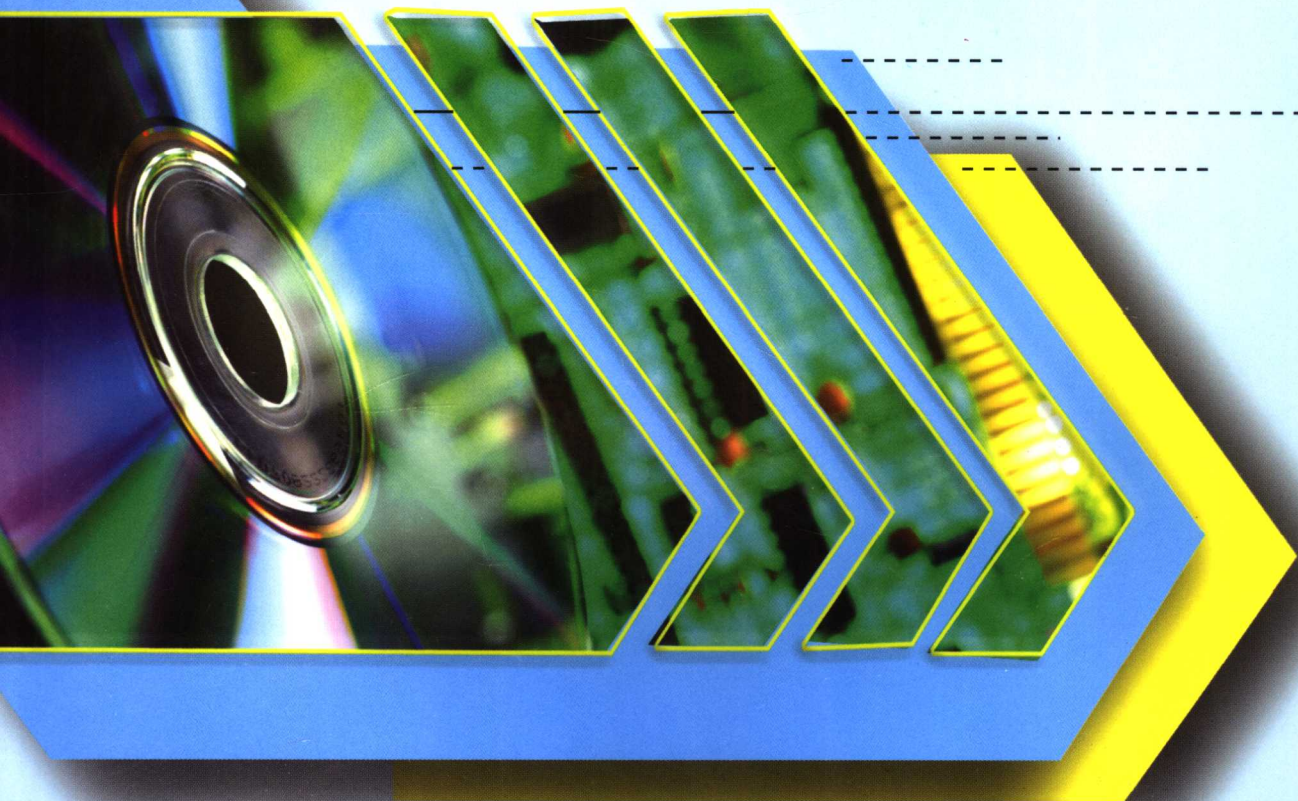


高等院校计算机应用技术规划教材
——实训教材系列
主编 谭浩强

C 程序设计 实训教程

崔武子 朱国瑛 白建明 编著



清华大学出版社

高等院校计算机应用技术规划教材

主 编 谭 浩 强

C 程序设计 实训教程

崔武子 朱国瑛 白建明 编著

清华大学出版社
北 京

内 容 简 介

本书以 C 程序设计零起点的读者作为主要对象,采用模板编写程序的方法,使学生快速入门,通过 86 个实训题目,学习所有知识点,再通过 83 个即学即练题目,巩固所学内容。

全书分 7 章,第 1 章介绍主函数的框架、C 语言数据类型和运算符等;第 2 章介绍 C 语言程序结构、标准库函数、自定义函数等;第 3 章介绍结构化程序设计的 3 种基本结构;第 4~7 章分别介绍指针、数组、结构体和文件。

本书精心制作的电子教案色彩协调,文字大小合适,动画丰富,完全可以脱离黑板讲授,也允许教师随意修改。该教案用链接的方式提供所有即学即练题目的代码和补充的调试程序题目,方便教师选用。

本书既可作为高职高专院校的教材,也可作为自学者的快速入门教材。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C 程序设计实训教程/崔武子,朱国瑛,白建明编著. —北京:清华大学出版社,2006.9

(高等院校计算机应用技术规划教材)

ISBN 7-302-13295-X

I. C… II. ①崔… ②朱… ③白… 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(2006)第 072454 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 装 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:16 字数:363 千字

版 次:2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

书 号:ISBN 7-302-13295-X/TP·8287

印 数:1~5000

定 价:22.00 元

编辑委员会

《高等院校计算机应用技术规划教材》

主 任 谭浩强
副 主 任 焦金生 陈 明 丁桂芝

委 员 王智广 孔令德 刘 星 刘荫铭
安志远 安淑芝 孙 慧 李文英
李叶紫 李 琳 李雁翎 宋 红
陈 强 邵丽萍 尚晓航 张 玲
侯冬梅 郝 玲 赵丰年 秦建中
莫治雄 袁 玫 訾秀玲 薛淑斌
谢树煜 谢 琛



《高等院校计算机应用技术规划教材》

进入 21 世纪,计算机成为人类常用的现代工具,每一个有文化的人都应当了解计算机,学会使用计算机来处理面临的事务。

学习计算机知识有两种不同的方法:一种是侧重知识的学法,从原理入手,注重理论和概念;另一种是侧重于应用,从实际入手,注重掌握其应用的方法和技能。不同的人应根据其具体情况选择不同的学习方法。对多数人来说,计算机是作为一种工具来使用的,应当以应用为目的,以应用为出发点。对于应用型人才来说,显然应当采用后一种学习方法。

传统的理论课程采用以下的三部曲:提出概念—解释概念—举例说明,这适合前面第一种侧重知识的学习方法。对于侧重应用的学习者,我们提倡新的三部曲:提出问题—解决问题—归纳分析。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。这种教学方法更适合于应用型人才。

应当指出,检查学习好坏的标准,不是“知道不知道”,而是“会用不会用”,学习的目的主要在于应用。因此希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有一些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而可以指定学生通过上机掌握这些内容。这样做可以培养学生的自学能力,启发学生的求知欲望。

1999 年我主编了一套《高职高专计算机系列教材》,由清华大学出版社出版,包括了 30 余种教材学习用书,受到全国各高校广大师生的欢迎,不仅许多高职高专学校选用它为教材,而且不少培养应用型人才的本科院校也认为很适合他们的教学特点,对提高教学质量起到了积极的作用。为了扩大这套教材的使用面,我们将这套教材改名为《高等院校计算机应用技术规划教材》,并对教材的内容进行了重新安排,包括 3 个子系列:

(1) 适合作为培养应用型人才的院校和基础较好、要求较高的高职高专学校的教材和参考书。封面颜色为黄色。

(2) 组织编写出版一批“实用教程”,内容主要是引导学生直接学习和掌



握计算机应用技能,对象是理论要求相对低一些或学时较少的学校(尤其是2年制高职高专)的需要。封面颜色为橘色。

(3) 组织若干种“实训教材”,其特点是侧重实践环节,引导学生通过自己的实践(而不是通过理论讲授)去获取知识,掌握应用。我们认为这应该是教学改革的一个重要方面,各类学校都可以选用这些实训教材。封面颜色为蓝色。

本套教材是针对培养应用型人才的高校和高职高专院校的教学特点组织编写的,包括了计算机专业和非计算机专业的教材和参考书。不同专业可以选择所需要的部分。本套教材包含的内容比较广,除了可作为正式教材外,还可作为某些专业的选修课或指定自学的教材。

本套教材是由一些普通高校和高职院校的老师编写的,他们对应用型的教学特点有较多的了解,也有较多的实践经验,保证了教材的质量。

由于我国的计算机应用技术教育正在蓬勃发展,许多问题有待深入讨论,新的经验也会层出不穷,本套教材的内容将会根据新的形势不断丰富和调整。

本套教材和参考书中肯定会有不足之处,请专家和读者不吝指正。

全国高等院校计算机基础教育研究会会长

谭浩强

2005年1月

前言



目前我国越来越重视高职高专的建设,现已有两年、三年和五年等高职教育模式,而且开设 C 程序设计课程的范围也越来越广。对于 C 程序设计课程,学生普遍反映枯燥、难学,不少教师也说每当讲授 C 程序设计课程时都要痛苦一次。那么本课程真的这么难学、难教吗?我们探索的教学模式的尝试结果并不是这样的。C 程序设计作为操作性极强的课程,让学生在实践与训练的过程中学习全部教学内容,效果很好。

本书是为初学程序设计语言的高职高专院校学生编写的。根据高职高专院校的培养目标和本课程的特点,我们在编写本书的过程中,特别注意快速入门,始终强调实践和应用。本书具有以下特点:

1. 通过 86 个实训题目,讲授所有教学内容

以强调实践能力为切入点,将 86 个实训题目分别放在 7 章介绍,并在完成实训题目的过程中,逐步介绍新的知识点。学生只要实践和理解这些题目,就可以基本掌握 C 语言了。此组织形式可在学生积极的配合下,系统地讲授程序设计的基础知识和编写应用程序的过程,达到节省较多课时、强调实践的目的。每道实训题目均按“问题提出—编程步骤(或编程分析)—程序代码—运行结果—技能要点—知识要点—即学即练—参考步骤”等顺序讲授。

对于教师来说,通过例题讲解新知识,要比介绍纯理论方便得多;

对于学生来说,理解具体实例,要比学习抽象的理论容易得多。

2. 通过 83 个即学即练题目,巩固所学内容

程序设计课程不能仅通过听来学会,还需要通过大量的实践。针对学生“读程序简单,独立编程难”的情况,本书在大部分实训题目后均安排了即学即练题目,并给出参考步骤。由于即学即练题目的知识点和对应实训题目相一致,教师可根据学生掌握情况和课时情况,选择部分或全部题目,要求学生在课内或课外完成。

3. 通过独特的电子教案,方便组织课堂教学

精心制作的电子教案色彩协调,文字大小合适,动画丰富,完全可以脱离黑板讲授,也允许教师随意修改。该教案用链接的方式提供所有即学即练题

目的代码和补充的调试程序题目,方便教师选用。该教案还包括全书的源代码。

4. 通过采用模板编写程序的方法,使学生快速入门

编写程序的过程分为先搭程序框架后填补代码的编程方式,以适应企业用人的特点。

5. 通过合理的内容顺序,遵循开发应用程序时的原则

在实际应用中,用C语言开发应用程序时,都是将一个较大任务分成若干小任务,通过编写相应的函数实现每一个小任务的功能。因此,本书遵循这一开发原则,从第2章开始学习和使用自定义函数。

6. 通过规范的格式,引导学生养成良好的编程习惯

较大的程序一般由多人协作完成,因此编写出来的程序不仅自己要看懂,还要让别人看懂。规范的缩进格式、见名知意的变量名都会提高程序的可读性。本书采用规范的缩进格式,大部分变量名见名知意,其中首字符表示变量数据类型。根据C语言定义的变量无确定值的缺点,定义变量时赋初始值。

7. 通过小结,归纳所学内容

每章通过增设小结的方式,提供本章中的重点、难点和扩展知识。

8. 通过贯穿整个教学过程的综合实训,帮助学生“瞻前顾后”

在每一章均提供一个贯穿整个教学过程的实训程序片段,随着讲授内容的增多,逐步补充和完善其程序的功能。此方法可使学生尽早接触应用程序的编写过程,也可使学生切实了解程序设计语言如何解决问题以及解决什么样的问题,进而更加明确这门课程的学习目标,增强对本课程的学习欲望。

9. 通过知识要点和独特的索引,方便查找所需内容

每章前列出了该章各实训题目的知识要点,附录中提供按关键字、运算符、库函数等分类的索引。

本书由崔武子主编、朱国瑛副主编,崔武子、朱国瑛、白建明共同编写,李红豫参与了部分内容的编写和程序的调试。

书中难免有错,请读者批评和指正。

作 者

2006年1月于北京

目录

第1章 C语言的基础知识	1
1.1 主函数的框架	1
1.2 熟悉 Turbo C 集成环境	3
1.3 正确选用 C 语言数据类型	7
1.4 使用 C 语言运算符	12
1.5 贯穿整个教学过程的综合实训部分程序	20
1.6 小结	21
训练题 1	26
第2章 函数	28
2.1 认识 C 语言程序结构	28
2.2 调用标准库函数	29
2.3 调用自定义函数	32
2.3.1 函数的定义	32
2.3.2 函数的调用	34
2.4 贯穿整个教学过程的综合实训部分程序	48
2.5 小结	51
训练题 2	52
第3章 3 种控制结构	54
3.1 结构化程序设计的 3 种基本结构	55
3.2 顺序结构的应用	55
3.3 分支结构的应用	58
3.3.1 if 语句与分支处理	58
3.3.2 switch 语句与分支处理	64

3.3.3 嵌套分支	70
3.4 循环结构的应用	74
3.4.1 while 语句与循环处理	75
3.4.2 do-while 语句与循环处理	84
3.4.3 for 语句与循环处理	88
3.4.4 嵌套循环	92
3.4.5 控制循环提前结束	98
3.5 贯穿整个教学过程的综合实训部分程序	102
3.6 小结	110
训练题 3	113
 第 4 章 指针	115
4.1 指针的概念	115
4.2 间接访问存储单元	116
4.3 指针的使用	119
4.4 贯穿整个教学过程的综合实训部分程序	124
4.5 小结	127
训练题 4	128
 第 5 章 数组	129
5.1 数组的定义与初始化	129
5.2 数组元素的引用	132
5.3 使用数组解决日常问题	135
5.4 字符串的处理	149
5.5 贯穿整个教学过程的综合实训部分程序	159
5.6 小结	171
训练题 5	172
 第 6 章 结构体	173
6.1 结构体类型的声明	173
6.2 结构体变量的定义和访问	174
6.3 结构体数组	184
6.4 结构体应用举例	186
6.5 贯穿整个教学过程的综合实训部分程序	192
6.6 小结	204

训练题 6	205
 第 7 章 文件	206
7.1 文件的概念	206
7.2 文本文件的基本操作	207
7.3 贯穿整个教学过程的综合实训部分程序	215
7.4 小结	229
训练题 7	229
 附录 I C 语言关键字	230
 附录 II 常用字符与 ASCII 代码对照表	231
 附录 III 运算符的优先级和结合方向	233
 附录 IV 常用 C 库函数	234
 附录 V 常见错误、警告信息表	238
 附录 VI 关键字、运算符、库函数索引	240
 参考文献	242

第1章

C 语言的基础知识

各实训知识要点:

- 实训 1.1 主函数的框架及 C 语言程序的初步认识。
- 实训 1.2 Turbo C 集成环境的使用。
- 实训 1.3 单步执行程序的方法。
- 实训 1.4 整型数据的取值范围;int 型和 long 型数据类型的合理选用。
- 实训 1.5 实型数据中的有效位数;float 型和 double 型数据类型的合理选用。
- 实训 1.6 字符常量的表示方法及存放形式;大小写字母的转换;转义字符。
- 实训 1.7 算术运算符、算术表达式及算术运算符的优先级的用法。
- 实训 1.8 强制类型转换符,求余运算符的使用要求。
- 实训 1.9 变量的初始化;自加、自减运算符。
- 实训 1.10 赋值运算符、复合运算符的使用。
- 实训 1.11 关系运算符的使用。
- 实训 1.12 逻辑运算符、各种运算符的优先级以及乘方的处理。

1.1 主函数的框架

【实训 1.1】 编写程序,在屏幕上先显示一串字符“Input two numbers:”,当用户输入两个数后,再显示这两个数和它们的和。

编程步骤:

- (1) 定义所需变量。
- (2) 输出“Input two numbers:”。
- (3) 输入两个数。
- (4) 计算两个数的和。
- (5) 输出所输入的两个数。
- (6) 输出两个数的和。

程序代码:

```
main()  
{  
    int a,b,sum;                /* 对应步骤(1) */  
  
    printf("Input two numbers:"); /* 对应步骤(2) */  
    scanf("%d%d",&a,&b);          /* 对应步骤(3) */  
    sum=a+b;                    /* 对应步骤(4) */  
    printf("The two numbers are %d and %d.\n",a,b); /* 对应步骤(5) */  
    printf("The sum is %d.\n",sum); /* 对应步骤(6) */  
}
```

运行结果:

```
Input two numbers: 3  5<回车>    (本书中用户输入的数据均用下划线表示)  
The two numbers are 3 and 5.  
The sum is 8.
```

技能要点:

(1) 任何一个 C 语言程序都包括 main() 函数(叫主函数), 程序中用虚线框标出来的部分是主函数的框架, 在大括号“{}”内编写具体操作的内容(叫做函数体)。

(2) 函数体由定义部分(空行上面的部分)和语句部分(空行下面的部分)组成, 而且所有定义部分必须都在语句部分前面, 有时根据函数的功能也可以没有定义部分。

(3) “int a,b,sum;”的作用是定义 3 个变量 a、b、sum。每个变量都代表内存中的一个存储单元, 使用变量前必须用定义变量的方式向系统申请空间。每个变量都要有自己的名字, 而且变量中的值可以随时改变。有关变量的详细介绍参见本章小结。

(4) C 语言不提供专门的输入、输出语句, 需要通过调用系统提供的标准库函数实现。本程序调用 scanf() 和 printf() 函数分别实现了输入、输出功能。

语句“printf(“Input two numbers:”);”的功能是在屏幕上输出字符串“Input two numbers:”, 其中“;”是语句的结束符, 不能丢。语句“printf(“The two numbers are %d and %d.\n”,a,b);”的功能是在输出双引号内的内容时, 用 a 和 b 的值从左到右依次代替两个 %d。其中“%d”是格式说明符, 在 scanf() 和 printf() 函数中控制输入、输出整数时使用; “\n”是换行符, 其作用是将输出位移动到下一行的第 1 列。

语句“scanf(“%d%d”,&a,&b);”的功能是将从键盘输入的两个整数分别赋给变量 a 和 b。有关 scanf() 和 printf() 函数的具体使用方法参见本章小结。

(5) 语句“sum=a+b;”的作用是从变量 a、b 所代表的存储单元中取出数据后计算它们的和, 并将其结果存放在变量 sum 所代表的存储单元中。

在程序中不能随意调换语句之间的顺序, 例如, 不能将“sum=a+b;”放在“scanf(“%d%d”,&a,&b);”之前。

(6) 用“/* */”表示的是注释部分, 它对程序的运行不起任何作用。在程序中适当

加注释,可以提高程序的可读性。由于开发好的软件过了很长时间后还需要阅读(例如,维护软件时),而且经常需要使别人看懂自己的代码(例如,共同完成软件的团队中的其他成员),所以在编写程序时应尽量多加注释。

(7) 编写任何程序之前必须先认真分析问题并确定解决问题的步骤,编完程序后通过运行程序的方法还要分析运行结果是否与预期结果相符。

知识要点:

①主函数的框架;②C 语言程序的初步认识。

1.2 熟悉 Turbo C 集成环境

【实训 1.2】 运行实训 1.1 程序,验证代码的正确性。

操作步骤:

(1) 启动 Turbo C 2.0 集成环境

在已安装 Turbo C 2.0 集成环境的情况下双击 TC.EXE 文件,将启动如图 1.1 所示的 Turbo C 2.0 集成环境。Turbo C 2.0 集成环境是集编辑、编译、连接、调试为一体的简单又实用的开发环境。如果没有安装该环境,将 Turbo C 2.0 直接复制到 C 盘 TC 文件夹中即可完成安装。

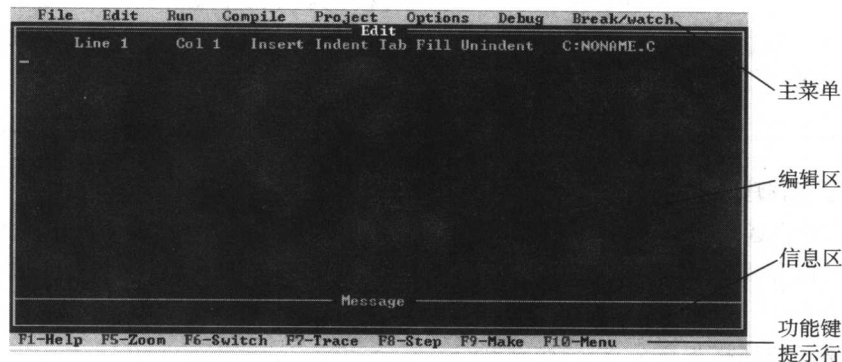


图 1.1 Turbo C 2.0 集成环境

在编辑区中输入或修改 C 语言源程序,在信息区中显示编译或连接程序时发现的错误位置和错误类型。

(2) 编辑程序

在编辑区中输入实训 1.1 中的程序。

```
main()
{   int a,b,sum;

    printf("Input two numbers:");
    scanf("%d%d",&a,&b);
```

```

sum=a+b;
printf("The two numbers are %d and %d.\n",a,b);
printf("The sum is %d.\n",sum);
}

```

输入程序时,编辑区中的 Line 和 Col 分别跟踪显示光标所在的行和列号。可通过键盘上的 Insert 键控制插入或改写状态。编辑(即输入或修改)源程序时,经常使用如表 1.1 所示的快捷键和块操作命令。

表 1.1 快捷键和块操作命令

命 令	功 能	说 明
F2	保存程序	当第 1 次保存时相当于“另存为”命令
F6	切换开关	实现编辑区和信息区之间的切换
F9	编译连接	运行程序之前通过编译和连接产生可执行文件
Ctrl+F9	运行程序	编译和连接后直接运行程序
F10	激活菜单	激活菜单后才能通过向左、向右光标键选择所需菜单
Alt+F5	查看结果	显示结果在 DOS 状态下屏幕中(黑色)
Alt+X	退出 TC	相当于 Windows 中的关闭按钮
Ctrl+KB	定义块首	把光标移动到块开始的位置后按此组合键
Ctrl+KK	定义块尾	把光标移动到块结束的位置后按此组合键
Ctrl+KV	移动块	把光标移动到块需要移动的位置后按此组合键
Ctrl+KC	复制块	把光标移动到块需要复制的位置后按此组合键
Ctrl+KY	删除块	删除块内的所有代码
Ctrl+KH	取消块	取消已定义的块

(3) 保存程序

按 F2 键或选择 File 菜单下的 Save 命令,在所出现的对话框中输入"d: \sx1_1<回车>",即可保存程序,文件的默认扩展名为“.c”。用 File 菜单下的 Write To 命令可更改文件名或路径。

(4) 编译和连接程序

编译和连接操作可按 F9 键进行。编译是将程序翻译成目标程序的过程(产生 sx1_1.obj),连接是将用户编写的程序和系统提供的库函数等连接起来产生可执行文件的过程(产生 sx1_1.exe)。如果编译或连接时有警告错误或语法错误,则在信息区中会显示出来,这时应修改错误后重新进行编译和连接。

(5) 运行程序

运行程序可按 Ctrl+F9 组合键进行。运行程序时也可能有错误,若发现错误,应修改后再一次进行编译和连接。实际上可以直接按 Ctrl+F9 键一次性进行编译、连接和运行操作,但这时只能发现语法错误,不能发现警告错误,因此建议先按 F9 键进行编译和连接程序后再按 Ctrl+F9 组合键运行程序。本题运行时需要输入两个整数,例如, 3 5<回车>。

(6) 查看运行结果并分析

运行结果显示在 DOS 屏幕,按 Alt+F5 组合键可以切换到 DOS 屏幕(由蓝色屏幕切换到黑色屏幕)。查看运行结果后,一定要判断所得到的运行结果与本问题的预期结果是否一致,如果不一致,则重复修改、编译、连接和运行操作,直到得出满意的结果为止。按任何键可从 DOS 屏幕回到 Turbo C 集成环境(由黑色屏幕切换到蓝色屏幕)。

技能要点:

(1) 在 Turbo C 环境编辑程序时要使用缩进格式,一般用 Tab 键实现缩进。按缩进格式输入一行后按回车键,光标自动在下一行的缩进位置上。例如,输入“sum=a+b;”后,光标自动在其首字符 s 的下面。

(2) 一定要养成先按 F9 键,对已编辑好的程序进行编译和连接后,再按 Ctrl+F9 组合键运行程序的好习惯,这一习惯使我们方便查找错误。例如,编辑下面程序后直接按 Ctrl+F9 组合键可以看到运行结果(如 a=3129),但其中 3129 的来历不清,说明程序肯定有问题。如果先进行编译和连接,则将发现如图 1.2 所示的警告错误,即对错误出现的行显示“给 a 赋值之前不能使用”的信息。

```
main()
{
    int a;
    printf("a=%d",a);
}
```

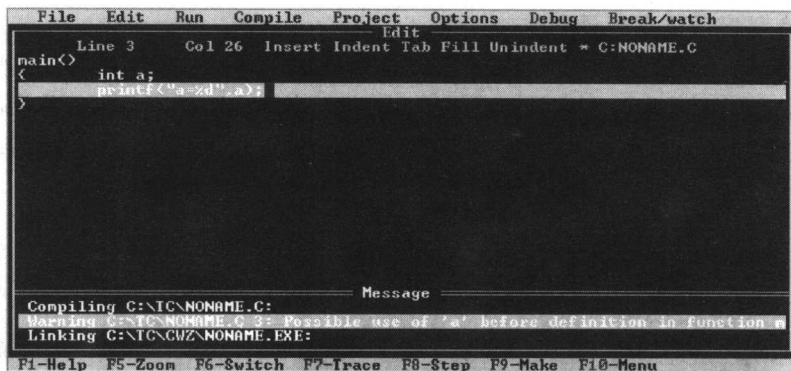


图 1.2 观察警告错误

(3) 经常按 F2 键保存程序。

(4) 只需验证代码的正确性时,可以不输入注释部分。

(5) 熟练操作,勤用快捷键。

知识要点:

Turbo C 集成环境的使用。

【实训 1.3】 直接在 Turbo C 环境中编写程序,根据输入的长、宽、高,计算长方体的体积,并上机一步一步观察程序中每一条语句的执行过程。

操作步骤:

(1) 先编写如下主函数的框架。

```
main()
{
}
```

(2) 补充函数体中的定义部分,需要定义存放长、宽、高、体积的 4 个变量。

```
main()
{   int a,b,c,v;
}
```

(3) 补充函数体中的语句部分,需要完成的操作是显示提示信息;输入 a、b、c 的值;计算体积;输出体积。完整代码如下:

```
main()
{   int a,b,c,v;
    printf("Input a,b,c:");
    scanf("%d%d%d",&a,&b,&c);
    v=a*b*c;
    printf("The volume is %d\n",v);
}
```

(4) 保存程序后进行编译和连接。

(5) 当程序中没有语法错误和警告错误时,按 F8 键开始单步执行程序。单步执行即为控制每行的执行。每当按一次 F8 键,程序就执行绿色条所在的语句行。在单步执行的过程中,可随时观察变量或表达式的值(用 Ctrl+F7 组合键设置监视项),也可以删除监视项,如图 1.3 所示。

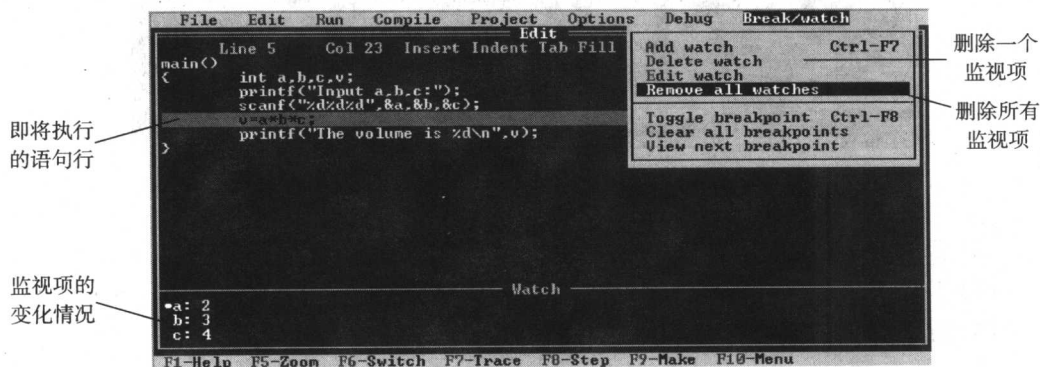


图 1.3 设置、观察和删除监视项

技能要点:

(1) 在入门阶段应学会模仿已有的程序编写类似的代码。

(2) 当程序的运行结果与预期结果不一样,而且错误出现的位置不易发现时,常使用