

21世纪高等学校规划教材

C语言程序设计 上机指导与习题

肖丽君 王冬梅 主编

21st Century University
Planned Textbooks



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校规划教材

C语言程序设计 上机指导与习题

肖丽君 王冬梅 主编

韩增红 佟继红 李明 孙淑霞 许盟 张泽梁 胡智鹏 孔令军 徐彬斌 编著

21st Century University
Planned Textbooks

人民邮电出版社

样书

专用章

人民邮电出版社

北京

图书在版编目 (CIP) 数据

C语言程序设计上机指导与习题 / 肖丽君, 王冬梅主编.
北京: 人民邮电出版社, 2009. 10
21世纪高等学校规划教材
ISBN 978-7-115-21366-2

I. C… II. ①肖…②王… III. C语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆CIP数据核字 (2009) 第162181号

内 容 提 要

本书是《C语言程序设计》的配套辅导用书, 主要介绍了C语言程序的编辑、编译、链接、运行和调试方法。实验包括验证性实验、程序分析类实验和设计性实验, 使读者在实验中掌握C语言程序设计的基础知识和应用。

本书分为三部分, 第一部分是上机实验, 共分12章, 由20个实验组成, 每个实验又包含多个实验项目, 内容丰富, 其中既包含实验范例, 又包含自编程序, 目的是采用循序渐进的方式, 提高读者分析问题和解决问题的能力; 第二三部分是习题与答案, 分11章, 每章习题都包括选择、判断、填空、程序改错和程序设计等题型, 并配有详细的解答。本书安排的内容具有很强的实用性和可操作性。

本书可作为高等院校各专业本、专科学生学习C语言程序设计的实验教程, 也可作为计算机培训和计算机等级考试的辅导用书, 还可作为广大程序开发人员和自学者的参考书。

21世纪高等学校规划教材

C语言程序设计上机指导与习题

-
- ◆ 主 编 肖丽君 王冬梅
 - 编 著 韩增红 佟继红 李 明 孙淑霞 许 盟
张泽梁 胡智鹏 孔令军 徐彬斌
 - 责任编辑 滑 玉
 - 执行编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 15.75
字数: 410千字
印数: 1—3 000册
 - 2009年10月第1版
2009年10月北京第1次印刷

ISBN 978-7-115-21366-2

定价: 26.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

目前,各高校理工类专业的计算机公共基础课大多开设了 C 语言程序设计课程,为使学生通过上机实验能够尽快地掌握 C 程序设计语言的语法规则,提高程序设计能力,达到事半功倍的效果,我们组织编写了《C 语言程序设计上机指导与习题》。

本书分三部分,第一部分是上机实验,分 12 章共有 20 个实验,每个实验包括实验目的、实验内容、实验步骤和思考与练习等内容。实验目的解决了“为什么做”的问题;实验内容解决了“做什么”的问题;实验步骤解决了“怎么做”的问题;思考与练习解决了“如何拓展实验内容”的问题。第二三部分是习题与答案,分 11 章,每章习题都包括选择、判断、填空、程序改错和程序设计等题型,并配有详细的解答。

本书上机实验部分的第 1 章详细介绍了 C 语言程序的编辑、编译、链接、运行和调试方法;并通过简单的实例说明如何使用 Turbo C 2.0 集成开发环境进行 C 语言程序设计,使读者了解 C 语言程序从编写到运行的全过程;第 2 章介绍了 C 语言程序设计的基础知识、数据类型、运算符和表达式的使用;第 3~5 章介绍了输入、输出函数的使用和结构化程序设计的 3 种基本结构,并以实例说明各流程控制语句的用法;第 6 章介绍了数组与字符串在程序设计中的应用;第 7 章介绍了函数的定义及应用、变量的存储类型和作用域以及程序编译预处理命令的使用;第 8 章介绍了结构体、共用体和枚举类型数据的使用;第 9 章介绍了指针的应用;第 10 章介绍了文件的读写和定位;第 11 章介绍了图形设计;第 12 章通过两个实例进一步训练读者解决实际问题的能力。习题部分读者可根据自身情况进行选做。

本书所使用的实例都经过反复挑选,既有实用性,又不乏趣味性,使学生在提高学习兴趣的同时,掌握相关的知识。书中全部例题和习题都上机调试通过。本书遵循由浅入深、循序渐进的原则,可作为高等院校本、专科院校高级语言程序设计课程的实验教材,也可作为广大计算机爱好者的自学读物。

本书由肖丽君、王冬梅主编。实验部分由韩增红编写第 1、7 章,王冬梅编写第 2 章,孙淑霞编写第 3 章,孔令军编写第 4 章,胡智鹏编写第 5 章,肖丽君编写第 6、10 章,佟继红编写第 8、11 章,李明编写第 9 章,张泽梁编写第 12 章;习题部分徐彬斌编写第 1~6 章,许盟编写第 7~12 章。

由于编者水平有限,书中难免有不足之处,恳请读者提出宝贵意见和建议。

编 者

2009 年 8 月

目 录

第一部分 上机实验

第 1 章 概述.....1	实验三 字符数组与字符串.....64
实验 C 语言程序的编辑与调试.....1	第 7 章 函数.....68
第 2 章 数据类型、运算符和 表达式.....11	实验一 函数的定义与调用.....68
实验一 数据类型、常量和变量.....11	实验二 变量的作用域.....77
实验二 运算符和表达式.....16	实验三 编译预处理.....83
第 3 章 顺序结构程序设计.....23	第 8 章 构造数据类型.....89
实验一 输入函数和输出函数.....23	实验 结构体、共用体与枚举类型.....89
实验二 顺序结构程序设计.....30	第 9 章 指针.....93
第 4 章 选择结构程序设计.....32	实验 指针.....93
实验 选择结构程序设计.....32	第 10 章 文件.....104
第 5 章 循环结构程序设计.....43	实验 文件的基本操作.....104
实验一 while 和 do-while 语句的使用.....43	第 11 章 图形设计.....111
实验二 for 循环语句的使用.....47	实验 简单图形设计.....111
第 6 章 数组.....53	第 12 章 综合应用.....116
实验一 一维数组.....53	实验一 计算器设计.....116
实验二 二维数组.....58	实验二 电话簿管理.....122

第二部分 习 题

第 1 章 概述.....135	第 5 章 循环结构程序设计.....156
第 2 章 数据类型、运算符和表达式.....137	第 6 章 数组.....168
第 3 章 顺序结构程序设计.....140	第 7 章 函数.....174
第 4 章 选择结构程序设计.....146	第 8 章 构造数据类型.....182

第 9 章 指针	187
第 10 章 文件	193

第 11 章 图形设计	198
-------------------	-----

第三部分 参 考 答 案

第 1 章 概述	203
第 2 章 数据类型、运算符和表达式	203
第 3 章 顺序结构程序设计	204
第 4 章 选择结构程序设计	206
第 5 章 循环结构程序设计	209
第 6 章 数组	214
第 7 章 函数	217

第 8 章 构造数据类型	221
第 9 章 指针	223
第 10 章 文件	225
第 11 章 图形设计	227

附录 Turbo C 编译错误信息	231
-------------------------	-----

参考文献	243
------------	-----

第一部分 上机实验

第 1 章 概述

实验 C 语言程序的编辑与调试

一、实验目的

1. 熟悉 Turbo C 2.0 集成环境。
2. 掌握 Turbo C 2.0 的启动与退出。
3. 掌握编辑运行 C 语言程序的基本步骤和方法。
4. 掌握 C 语言程序的调试方法。
5. 通过运行调试简单的 C 程序，初步了解 C 语言程序的特点。

二、实验内容

1. 查看 Turbo C 2.0 的系统组成。

Turbo C 2.0 系统主要包含以下文件：

INSTALL.EXE	安装程序文件
TC.EXE	集成编译
TCINST.EXE	集成开发环境的配置设置程序
TCHELP.TCH	帮助文件
THELP.COM	读取 TCHHELP.TCH 的驻留程序
README	关于 Turbo C 的信息文件
TCCONFIG.EXE	配置文件转换程序
MAKE.EXE	项目管理工具
TCC.EXE	命令行编译
TLINK.EXE	Turbo C 系列连接器
TLIB.EXE	Turbo C 系列库管理工具
C0?.OBJ	不同模式启动代码

C?.LIB	不同模式运行库
GRAPHICS.LIB	图形库
EMU.LIB	8087 仿真库
FP87.LIB	8087 库
*.H	Turbo C 头文件
*.BGI	不同显示器图形驱动程序
*.C	Turbo C 例行程序（源文件）

其中，上面的“？”分别为：

T	Tiny（微型模式）
S	Small（小模式）
C	Compact（紧凑模式）
M	Medium（中型模式）
L	Large（大型模式）
H	Huge（巨型模式）

2. 启动 Turbo C 2.0 系统，熟悉其集成环境。
3. Turbo C 2.0 系统的菜单操作。
4. 退出 Turbo C 2.0 系统。
5. 创建一个简单的 C 语言程序，进行编辑保存、编译连接、调试运行并查看运行结果。
6. C 语言程序的调试。

程序调试是指对程序的查错和排错。一般来说，程序的错误包括语法错误和逻辑错误。语法错误在程序编译的过程中会被发现，并在编译后由编译器给出错误提示；逻辑错误是因为解决问题的方法不正确造成的，编译系统无法发现错误，但可以根据程序运行的结果是否符合要求来判断程序中是否存在逻辑错误。

程序调试主要有两种方法：静态调试和动态调试。调试程序一般需要经过以下几个步骤。

（1）进行人工检查，即静态调试。

在程序编写完之后，不要匆匆忙忙上机，而应先对程序进行人工检查。这一步十分重要，它能发现程序设计人员由于疏忽而造成的多数错误。通过静态检查，可以大大缩短上机调试的时间，提高上机效率。

静态调试主要是对程序的语法规则和逻辑结构进行检查。检查程序语句、语法的正确性；检查程序中各变量的初值和初值位置的正确性；检查程序中选择结构的正确性；检查程序中循环结构的循环次数和循环嵌套的正确性；检查表达式的合理性。

程序的静态调试是程序调试非常重要的一步。初学者应培养自己静态调试的良好习惯，在上机前认真做好程序的静态调试工作。

（2）在人工检查无误后，再上机调试，即动态调试。动态调试贯穿于编译、连接和运行程序的整个过程中。

① 编译过程中的调试。

在编译时会给出语法错误的信息，调试时可以根据提示信息找出程序中具体出错之处并改正。应当注意的是系统指明的出错位置未必就是真正出错的位置，通常情况是错误出现在前，而系统发现错误在后，因为它检查到某个地方发生实际错误之后，才能确认出了问题，并报出错误信息。要确认第一个错误的原因，应该从系统指明的位置开始进行检查，并从此处开始向前检查。如果

系统提示的出错信息很多，应当从上到下逐一改正。有时显示出一大片出错信息往往使人感到问题严重，无从下手。其实可能只有一两个错误。例如，对使用的变量未定义，编译时就会对所有含该变量的语句发出出错信息。这时只要加上一个变量定义，所有错误就都消除了。

有时提示出错的类型并非绝对准确，由于出错的情况繁多且各种错误互有关联，因此要善于分析，找出真正的错误，而不要只从字面意义上找出错原因。

② 连接过程中的调试。

在改正语法错误（包括“error（错误）”和“warning（警告）”）后，程序经过连接就得到可执行的目标程序。连接过程有时也会出错，一般的错误有：找不到某个库函数或某个库文件；某个外部调用不正确；某些模块的参数超过系统限制等。例如，在使用 `printf()` 函数时，写成了“print”，那么在连接此库函数时，就会因找不到该库函数而报错。

③ 运行过程中的调试。

运行程序，输入程序所需数据，即可得到运行结果。应当对运行结果作出分析，看它是否符合要求。

有时，数据比较复杂，难以立即判断结果是否正确。可以事先考虑好一批“试验数据”，输入这些数据可以很容易判断结果正确与否。例如，if 语句有两个分支，有可能程序在经过其中一个分支时结果正确，而经过另一个分支时结果不对，必须考虑周全。事实上，当程序复杂时很难把所有的可能情况全部都试到，可以选择典型的临界数据作试验。

运行时的错误大体分两类：一类是与数据的输入、输出格式有关，一般容易发现这类错误，例如，要输入长整型（类型格式说明符是 `%ld`）数据，却把类型格式说明符写成了“`%d`”；另一类属于逻辑错误，对这类错误往往需要仔细检查和分析才能发现。

通过上机调试程序，掌握程序调试的方法和技术，学会自己找问题，这样慢慢就会写出错误较少的实用程序。

三、实验步骤

1. 查看 Turbo C 2.0 的系统组成。

在 Windows 资源管理器中打开 Turbo C 2.0 系统所在的目录，对照上述 Turbo C 2.0 系统主要包含的文件，了解 Turbo C 2.0 系统的组成。

2. 启动 Turbo C 2.0 系统，熟悉其集成环境。

安装完毕的 Turbo C 2.0 将在 C 盘根目录下建立一个 TC 子目录，TC 下还建立了两个子目录 LIB 和 INCLUDE，LIB 子目录中存放库文件，INCLUDE 子目录中存放所有的头文件。

Turbo C 2.0 的启动方法有以下几种，Turbo C 2.0 的集成化操作界面如图 1-1 所示。

（1）在 DOS 系统界面下运行 Turbo C 2.0 时，首先要进入 TC 子目录，然后键入“TC”，并按 Enter 键，即可启动 Turbo C 2.0 集成开发环境。

（2）在 Windows 系统中运行 Turbo C 2.0 时，首先单击“开始”按钮，在弹出的开始菜单中选择“所有程序|附件|命令提示符”，进入 DOS 系统界面；然后，键入“`cd\tc`”，并按 Enter 键，进入 TC 子目录；再键入“TC”，并按 Enter 键，即可启动 Turbo C 2.0 集成开发环境。

（3）在 Windows 资源管理器中打开 Turbo C 2.0 系统所在的目录，双击 TC.EXE 的图标，即可启动 Turbo C 2.0 集成开发环境。

（4）如果 Windows 桌面上已建有 Turbo C 2.0 的快捷方式，双击快捷方式图标，即可启动 Turbo C 2.0 集成开发环境。

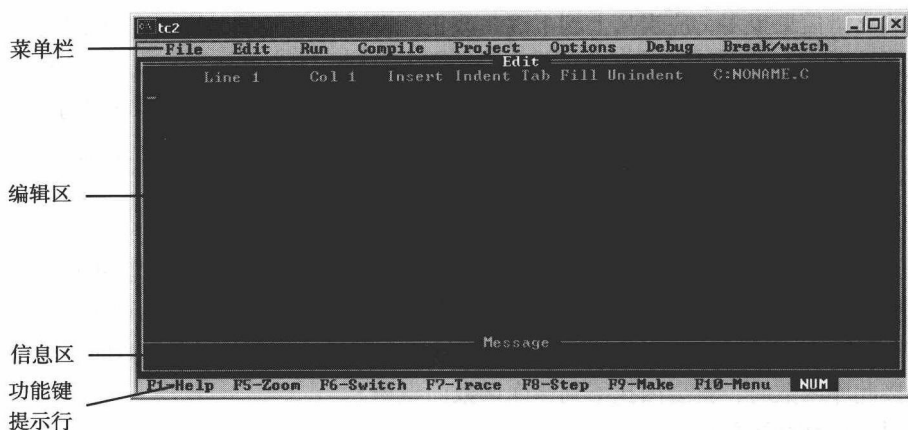


图 1-1 Turbo C 2.0 的集成化操作界面

对已打开的 Turbo C 2.0 窗口，可以按 Alt+Enter 组合键，在窗口方式和全屏方式之间进行切换。

3. Turbo C 2.0 系统的菜单操作。

(1) 进入 Turbo C 2.0 集成环境后，若要使用主菜单，有以下两种方法。

① 按 Alt+主菜单的第一个字母键打开主菜单项，进入该项的子菜单中，若要取消菜单，按 Esc 键即可。

② 按 F10 键激活菜单栏，然后用键盘上的左、右箭头键选择某项主菜单，并按 Enter 键打开此项主菜单，进入该项的子菜单中。

(2) 要选择子菜单，按键盘上的上、下箭头键选择某项子菜单，然后按 Enter 键启用该项子菜单。

(3) 对于有热键的功能菜单，也可以使用热键启用它。例如，可用热键的方式完成一个程序的编辑、编译、连接和运行：按 F3 键，可在随之出现的对话框中输入文件名，文件名可以带“.c”，也可以不带（此时系统会自动加上）；输入文件名后，按 Enter 键，可打开该文件，如果该文件不存在，就建立一个新文件，系统随之进入编辑状态；此时就可以输入或修改源程序了，然后按 Ctrl+F9 组合键，则进行编译、连接和执行，这 3 项工作是连续完成的。

4. 退出 Turbo C 2.0 系统。

选择 File 菜单中的 Quit 选项，或按 Alt+X 键，返回到 DOS 状态。若要返回到 Windows 系统，可键入“EXIT”，然后按 Enter 键。

5. 创建一个简单的 C 语言程序，进行编辑保存、编译连接、调试运行并查看运行结果。

通常建立并执行一个 C 程序有以下几个步骤。

(1) 在编辑器中编写源文件。

(2) 编译。

(3) 连接。

(4) 运行可执行文件。

建立一个名为“mycfile.c”的 C 语言源程序，熟悉在 Turbo C 集成开发环境中建立一个新程序的步骤。

(1) 编辑源程序文件。在 Turbo C 的编辑区中输入如下程序，如图 1-2 所示。

```
void main()                /* 主函数 */
{
```

```

printf("Hello, world!\n");    /* 输出语句 */
printf("I am a student.");    /* 输出语句 */
}

```

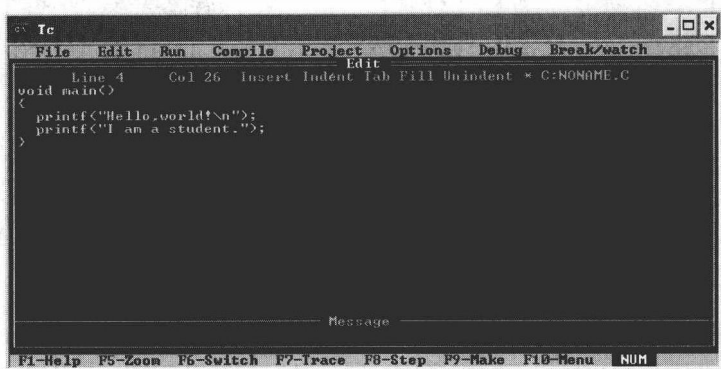


图 1-2 在 Turbo C 中输入源程序

(2) 保存源程序文件。按 Alt+F 组合键打开 File 菜单，选择 Save 命令存盘，或者直接按 F2 键将文件存盘。这时，系统将弹出一个 Rename NONAME 对话框，输入文件名 “mycfile.c”，并按 Enter 键，如图 1-3 所示。

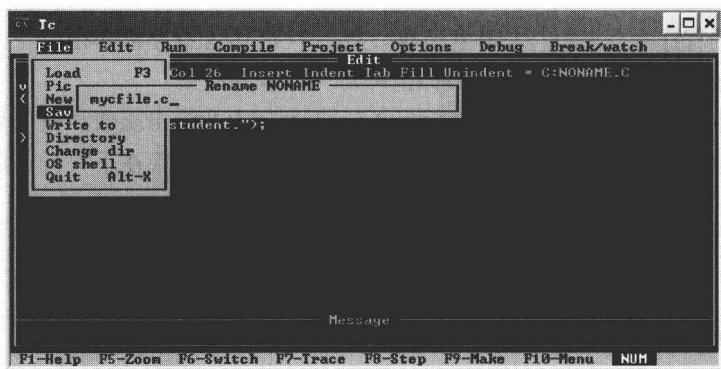


图 1-3 保存源程序文件

(3) 编译源程序。按 Alt+C 组合键打开 Compile 菜单，选择 Compile to OBJ 命令进行编译，如图 1-4 所示；或者按 Alt+F9 组合键进入编译状态，这时，系统弹出一个 Compiling（编译）对话框，若显示信息 “Success: press any key”，则表示编译成功；如果编译时产生 Warning（警告）或 Error（错误）信息，会显示在屏幕下部的 Message（信息）窗口中，可按提示对源程序进行修改，并重新进行编译。编译成功后，系统会生成一个与源程序文件同名的目标程序文件 mycfile.obj。

(4) 连接程序。按 Alt+C 组合键打开 Compile 菜单，选择 Link EXE file 命令，或者按 F9 键进入连接状态，系统将各目标程序与有关的库文件进行连接，生成一个可执行文件。与编译时一样，系统弹出一个 Linking（连接）对话框，若显示信息 “Success: press any key”，则表示编译成功；如果编译时产生 Warning（警告）或 Error（错误）信息，会显示在屏幕下部的 Message（信息）窗口中，可按提示对源程序进行修改，并重新进行编译连接。连接成功后，系统会生成一个与源程序文件同名的可执行文件 mycfile.exe。

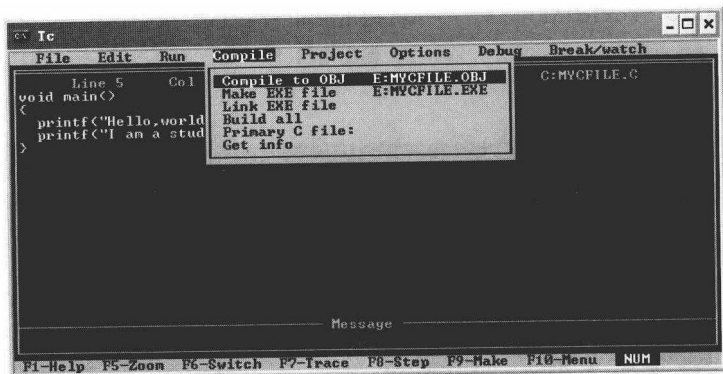


图 1-4 编译操作

(5) 运行程序。源程序经编译连接无误后, 按 Alt+R 组合键打开 Run 菜单, 选择 Run 命令, 或者按 Ctrl+F9 组合键运行程序, 运行完毕后, 屏幕又回到编辑窗口。打开 Run 菜单, 选择 User Screen 命令, 或者按 Alt+F5 组合键切换到程序输出窗口, 才能查看到输出结果。再按任意键, 又回到编辑窗口。

上述运行程序的方法是将程序的编译、连接和运行 3 个操作连在一起完成的。源程序编辑保存后, 也可直接运行。

程序分析

(1) 在上面的程序中, main() 表示主函数, 每一个 C 语言程序都必须有一个主函数 main()。void 表示主函数 main() 没有返回值。函数体由一对大括号括起来。

(2) 本例中主函数有两条输出语句, printf 是 C 语言中的输出函数。双引号内的字符串按原样输出, “\n” 是换行符, 即在输出 “Hello, world!” 后, 光标出现在下一行, 语句的最后必须有一分号, 分号是 C 语言语句的结束标志。

(3) /* 和 */ 括起来的是注释部分, 将有助于程序的阅读和理解, 要注意 “/” 和 “*” 之间不能有空格。

程序运行结果

```
Hello, world!
I am a student.
```

6. C 语言程序的调试。

程序的编译和连接没有错误不等于运行时一定正确。编译系统能检查出语法错误, 但无法检查出逻辑错误。下面介绍两种程序动态调试的方法。

(1) 单步法: 程序每次执行一行, 用户可检查此时各有关变量和表达式的值, 以便发现错误。主要用到的功能键如下。

F7 键: 按步执行程序, 每按一次, 执行一条语句。

Ctrl+F7 组合键: 观察变量和表达式的值。

(2) 断点法: 程序执行到断点时将暂停, 用户可检查此时各有关变量和表达式的值, 如果未发现错误, 就使程序继续执行到下一断点, 如此逐段检查, 就能不断地缩小出错的范围, 直到找到出错点。主要用到的功能键如下。

Ctrl+F8 组合键: 设置/取消断点。

Ctrl+F4 组合键: 查看修改变量的值。

通过以下实例，说明两种调试方法的使用。

(1) 打开 Turbo C 2.0 集成环境，输入如下程序。该程序的功能是从键盘上输入两个整数，求其中较大者。在主函数 `main()` 中从键盘接收两个整数，通过自定义子函数 `max()` 来求得两者的最大值，并在主函数 `main()` 中输出。

```
int max(int x,int y)
{
    if(x>y)
        return y;
    else
        return x;
}
void main()
{
    int a,b,maxNum;
    printf("Input two integer:");
    scanf("%d%d",&a,&b);
    maxNum=max(a,b);
    printf("The max is: %d\n",maxNum);
}
```

(2) 单步执行调试程序。

根据前述方法进行编辑、编译、连接和运行这个程序，可以发现这个程序没有任何语法错误，能顺利通过编译和连接。在程序运行时，屏幕出现提示 “Input two integer:”，输入 8 和 5，并按 Enter 键；程序运行的结果显示：

The max is:5

如图 1-5 所示，这个结果显然是不对的。这时可采用如下所述的单步执行程序的方法，检查每一步的正确性。

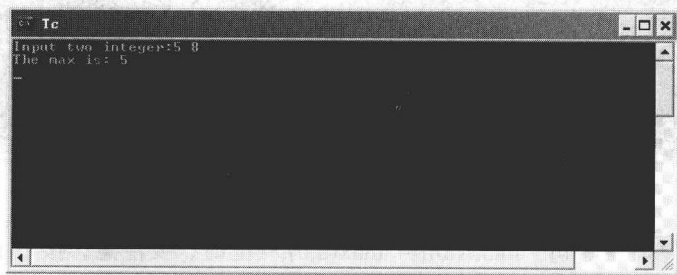


图 1-5 查看程序运行结果

① 重新开始运行，按 F7 键，可以看到编辑窗口中源程序的主函数 `main()` 处高亮度显示，表示准备进入主函数 `main()`。同时可以看到屏幕下部的 Message 窗口变成了 Watch 窗口，它是用于观察数据的，如图 1-6 所示。

② 再次按下 F7 键，高亮条移到主函数中的第 2 行（第 1 行是对变量的定义，不是执行语句，故被跳过），表示已进入了主函数 `main()` 和左花括号，但并未执行 `printf` 函数语句，只是表明下一步要执行此语句。

③ 按 F7 键，此时执行主函数的第 2 行语句，在用户屏幕上将输出 “Input two integer:” 的信息。按 Alt+F5 组合键可以查看用户屏幕，然后按 Esc 键返回编辑窗口。

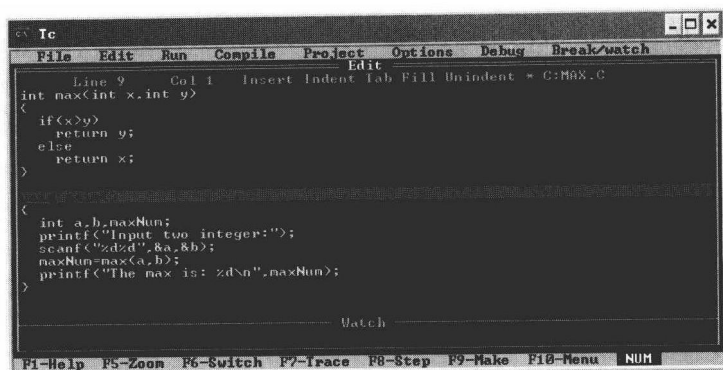


图 1-6 用 F7 键跟踪的初始窗口

④ 按 F7 键，此时执行主函数的第 3 行语句，由于 scanf 函数语句需要输入数据，所以程序将切换到用户屏幕，输入：

8 5

按 Enter 键后，屏幕返回编辑窗口，高亮条移到下一行，表示 scanf 语句已执行完毕。这时，变量 a 已被赋值为 8，变量 b 已被赋值为 5。查看变量的方法是按 Ctrl+F7 组合键，将弹出一个 Add Watch 对话框，输入想查看的变量名 a，按 Enter 键后，该对话框消失，在屏幕下部的 Watch 窗口显示变量 a 的当前值为 8，如图 1-7 所示。按同样的方法在 Watch 窗口中添加变量 b 的值。

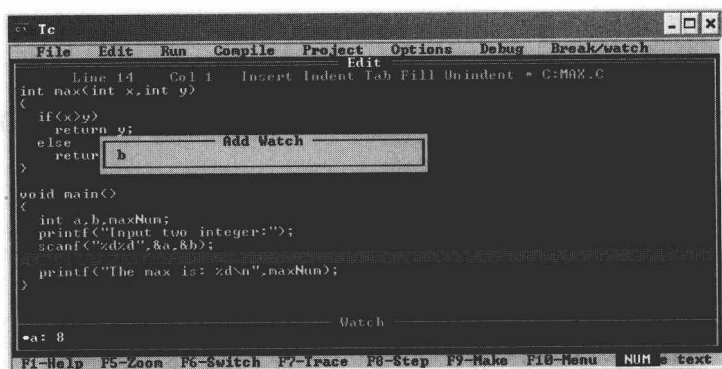


图 1-7 添加变量到 Watch 窗口

⑤ 当高亮度条停留在语句 “maxNum = max(a,b);” 上时，要注意，按下 F7 键后，程序将转去执行子函数 max()，即高亮度条将移到子函数 max() 上。

这里要注意，与功能键 F7 相似的 F8 键也是单步执行程序，但 F8 键在执行有函数的语句行时，不跟踪到函数内部。即若在此行按下 F8 键，则程序不进入函数 max() 中，而是将高亮条移到下一句的 printf 语句上。

⑥ 按 F7 键高亮条移到 if 语句上，判断 x 和 y 大小。

⑦ 按 F7 键高亮条移到 “return y;” 语句上，即若 x 的值大于 y 的值，则返回的是 y 的值。这与函数的功能相悖。因此知道程序出错的原因是子函数 max() 有问题，从而返回了错误的值。这时可按 Ctrl+F2 组合键中断程序的执行，并且将子函数 max() 改为：

```
int max(int x,int y)
{
```

```

if(x>y)
    return x;          /* 将返回 y 的值改为返回 x 的值 */
else
    return y;          /* 将返回 x 的值改为返回 y 的值 */
}

```

重新按 Ctrl+F9 组合键运行程序，得到正确的结果为：

```

Input two integer: 8 5
The max is: 8

```

(3) 用设置断点的方法调试程序。

① 将程序恢复为修改前。为了找出问题所在，可在程序中主函数 main() 中的函数调用语句 “maxNum=max(a,b);” 行上设置断点，即把光标移到此行上，并按 Ctrl+F8 组合键，此行即被红色条覆盖，如图 1-8 所示。

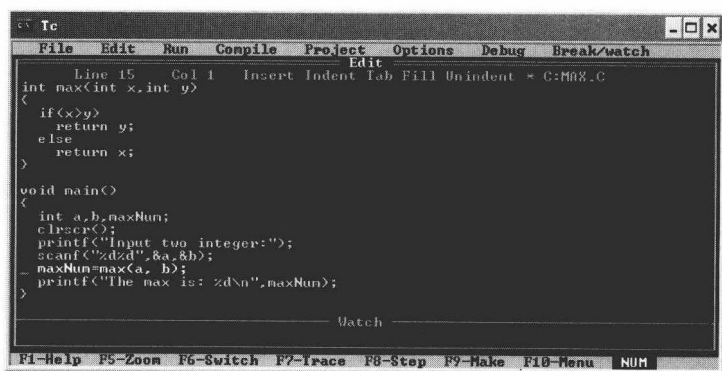


图 1-8 设置断点的窗口

在此行加入断点的作用是可透过 Watch 窗口观察变量 a 和 b 在调用函数 max() 之前是否正确，还可以观察设置断点的行执行后所得到的变量 maxNum 的值是否正确。

② 按 Ctrl+F9 组合键运行程序，执行到 scanf 函数时，输入 8 和 5。然后程序执行到断点行暂停。应注意的是，此时的断点行并未执行。这时我们可以通过前面讲述的 Watch 窗口来查看变量 a 和变量 b，如图 1-9 所示。

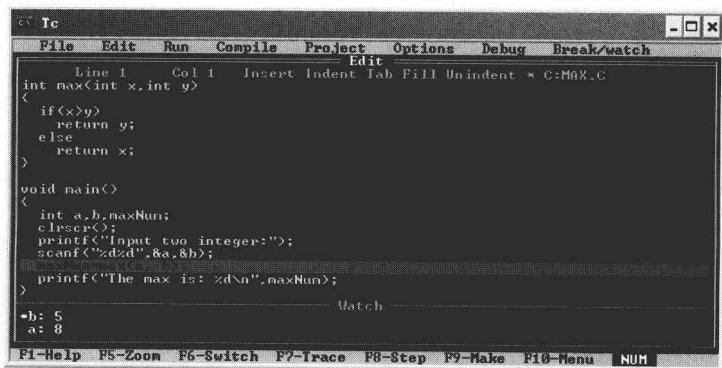


图 1-9 设置断点的跟踪窗口

在图 1-9 中，可以看到变量 a 和变量 b 的值都是正确的。即传入函数 max 的参数是正确的，

那么一定是函数 `max()` 发生了错误。

③ 按 F7 键, 跟踪程序进入函数 `max()`, 可以发现如前所述的错误原因。改正错误后, 在断点行按 Ctrl+F8 组合键取消所设置的断点, 按 Ctrl+F9 组合键重新运行, 则程序得到正确的结果。

四、思考与练习

1. 什么是语法错误? 什么是逻辑错误?
2. 编写一个程序, 利用 `printf()` 函数, 在第一行打印自己的姓名, 第二行打印自己的地址。
3. 按照程序调试步骤上机运行以下程序。程序的功能是: 已知 5 个学生的数学成绩分别为 87、69、92、75、97, 计算并输出这 5 个学生的平均成绩 (程序运行结果为: `average = 84.000000`)。

```
void main()
{
    int a,b,c,d,e;
    float sum,aver;
    a=87; b=69; c=92;
    d=75; e=97;
    sum=a+b+c+d+e;
    aver=sum/5;
    printf("average=%f\n",aver);
}
```

4. 打开上题中的程序, 并作以下修改:

- (1) 将第 1 行中的 `main` 改为 `Main`。
- (2) 将第 2 行最后的分号 “;” 去掉。
- (3) 将第 4 行的 3 条赋值语句都删除, 改为 `printf("Input 5 integers: ");`。
- (4) 将第 5 行的 2 条赋值语句都删除, 改为 `scanf("%d,%d,%d,%d,%d",&a,&b,&c,&d,&e);`。

运行改变后的程序, 若出现错误, 请观察其错误信息, 并调试程序, 改正错误。

程序修改正确后, 运行时输入与上题相同的 5 个数据: 87,69,92,75,97。观察运行结果, 与上题中程序的运行结果进行比较。

第 2 章

数据类型、运算符和表达式

实验一 数据类型、常量和变量

一、实验目的

1. 掌握 C 语言的各种数据类型。
2. 掌握 C 语言中常量的定义和使用方法。
3. 掌握 C 语言中变量的定义和使用方法。

二、实验内容

1. 常量的使用：将整型常量 5 赋给变量 a，实型常量 3.5 赋给变量 b。
2. 编写程序：求半径为 3.5 的圆的周长和面积。
3. 变量要先定义，后使用。

输入如下程序，分析运行结果。

```
main()
{
    a=2;b=3;
    sum=a+b;
    printf("%d",sum);
}
```

4. 整型和实型数值溢出问题。

输入如下程序，分析运行结果。

```
main()
{ int a;
  float b;
  a=50000;
  b=1e39;
  printf("%d,%e",a,b);
}
```

5. 大小写字母的转换。

输入如下程序，分析运行结果。

```
main()
{ char a;
```