



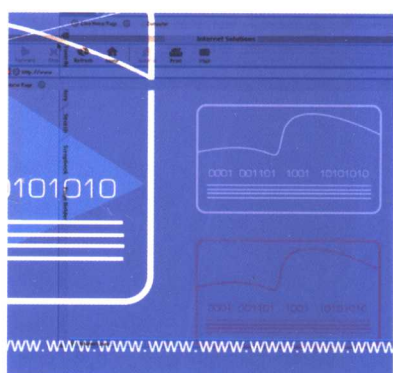
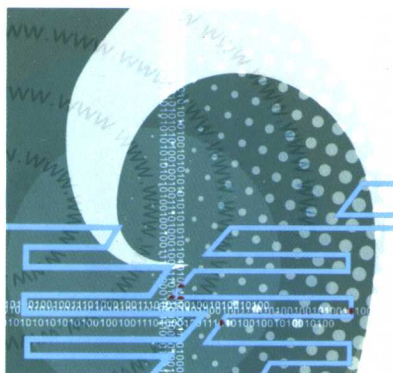
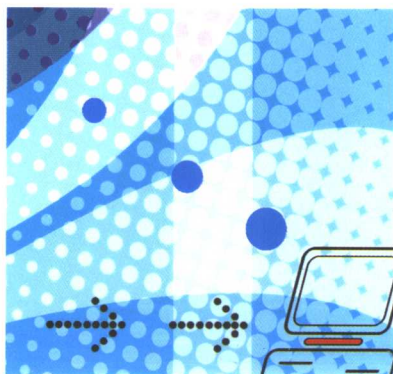
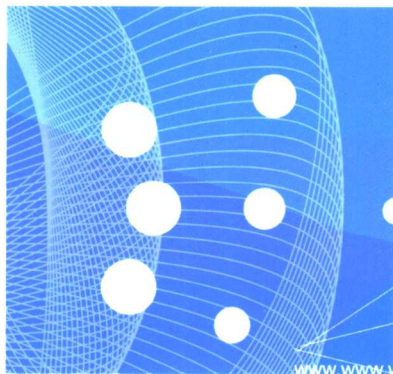
21 世纪高等学校应用型教材

C 程序设计教程

上机辅导与习题集

□ 谭浩强 主 编

□ 卜家岐 范燮昌 编著



高等教育出版社
Higher Education Press

21 世纪高等学校应用型教材

C 程序设计教程 上机辅导与习题集

谭浩强 主编

卜家岐 范燮昌 编著

高等教育出版社

内容简介

本书是谭浩强主编的《C 程序设计教程》(简称“主教材”)的配套教材,侧重对 C 程序设计上机练习的辅导。全书共 5 章,主要内容包括: Turbo C 2.0 集成开发环境介绍、用 Visual C++ 6.0 开发 C 程序、主教材习题参考解答、其他典型题解和二级等级考试部分题型求解方法。

本书适合作为大学本科、高职高专、成人高校和其他初学者学习 C 程序设计的辅导教材,也可供参加全国计算机等级考试(二级 C)的各类读者选用。

图书在版编目(CIP)数据

C 程序设计教程上机辅导与习题集 / 谭浩强主编; 卜家岐, 范燮昌编著. —北京: 高等教育出版社, 2006.1

ISBN 7-04-018144-4

I. C... II. ①谭... ②卜...③范... 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 (2005) 第 143574 号

策划编辑	雷顺加	责任编辑	萧 潇	封面设计	王凌波	责任绘图	黄建英
版式设计	胡志萍	责任校对	朱惠芳	责任印制	宋克学		

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京凌奇印刷有限责任公司

开 本 787×1092 1/16
印 张 8.75
字 数 210 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2006 年 1 月第 1 版
印 次 2006 年 1 月第 1 次印刷
定 价 11.60 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 18144-00

前 言

“C 程序设计”是一门实践性很强的课程，只有通过多编程、多上机、多实践才能真正掌握该课程的内容。在进行上机实验前，一定要有充分的准备，对需求解的问题要有一个初步的算法。这个算法不要只停留在脑子里，最好是用常用流程图或 N-S 流程图的形式表达出来，这在主教材中已有举例说明。另外还可以用自然语言（中英文都可以）和计算机语言中的几种语句结构相结合来描述算法，即用伪代码（pseudo code）来描述算法，这比画流程图更直接、省力，并且更容易转换为程序，根据伪代码算法写出程序初稿即可上机调试。例如，要求 100~1 000 范围内的水仙花数（一个三位数的个位数、十位数和百位数的立方之和等于该数本身），用伪代码描述算法可以写成如下形式：

```
定义变量 n,i,j,k
for(n=100;n<=1000;n++)
{
    取出 n 的百位数 i
    取出 n 的十位数 j
    取出 n 的个位数 k
    if(n 等于 i、j、k 的立方和)
        打印 n 等于 i、j、k 的立方和
}
```

程序调试可分两个阶段：首先纠正程序中的语法错误；然后纠正程序中的算法错误。

1. 纠正程序中的语法错误

纠正程序中的语法错误，即纠正程序中不符合 C 语言所规定的各种书写方式的错误，这可以通过编译系统的自动检查来获得相关错误的信息。通过编译，若程序有语法错误，系统会给出全部出错信息和错误所在的位置，便于设计者修改错误。有时还需要通过多次编译和纠错才能纠正程序中的全部错误。值得注意的是，信息窗口中列出的全部错误中的第一条错误，必须认真对待并及时修正，因为其后的错误可能是由于第一条错误引起的。经过几次纠错和编译后，错误会越来越少，直至纠正所有的错误。

2. 纠正程序中的算法错误

算法错误又称程序逻辑错误或运行结果错误，是指程序虽然可以运行，但得不到正确的结果，这就要重新修改原来的算法，根据修改后的算法修改程序，重新编译连接和运行。正确的算法得到正确的结果，不正确的算法得到不正确的结果，这只能靠编程者自身对求解问题的理解程度来决定，计算机是无能为力的。一般情况下，系统不会给出算法错误的相关提示信息，这就需要设计者利用开发环境提供的调试手段，逐步找出程序设计不合理的地方。从这一点看，调试程序时的第二个阶段比第一个阶段难度要大得多，但对于一些解决简单问题的程序的调试，

多数只经历第一个阶段，而不一定会出现第二个阶段。

本书对程序调试的这两个阶段都用具体例子进行了较详细的说明。

本书第1章介绍 Turbo C 2.0 集成开发环境和一些调试程序的基本方法；第2章介绍用 Visual C++ 6.0 开发 C 程序；第3章是主教材中每章习题的参考解答；第4章介绍其他典型题解；第5章介绍等级考试（二级）中一些常见题型的求解方法。

本书由全国高校计算机基础教学研究会理事长谭浩强教授主编，主要由上海大学卜家岐教授和范燮昌副教授执笔编写。参加本书大纲讨论、程序调试和部分内容编写工作的还有陈章进、侯庆祥、康惠骏、邹启明、王晓冬、罗真、徐安东、陆黎明、黄俊民、余慧蔼等。作者都具有丰富的 C 语言教学经验和软件设计的实践经验。初稿完成后，又由三位多年从事 C 语言一线教学的主讲教师（北京科技大学的徐新华副教授、同济大学高等技术学院的程克明副教授和承德石油高等专科学校的马晓晨副教授）进行了认真的审读，他们提出了不少宝贵意见。根据审读者所提意见，作者对原稿又做了认真的研究、讨论和修改，使本书的适用性进一步提高，在此对他们表示真挚的谢意。

由于作者水平有限，加之时间仓促，书中难免有错漏之处，敬请读者不吝指正。

作者

2005 年 10 月

目 录

第 1 章 Turbo C 2.0 集成开发环境介绍.....1	练习十..... 78
1.1 概述.....1	第 4 章 其他典型习题解答..... 84
1.2 使用 TC 的主要步骤.....2	4.1 多任务求值计算..... 84
1.3 修正编译过程中的语法错误.....5	4.1.1 综合法..... 84
1.4 修正程序的算法错误.....6	4.1.2 分程序法..... 85
1.5 含命令行参数程序的调试.....12	4.1.3 过程函数法..... 86
1.6 TC 集成开发环境的配置文件.....13	4.1.4 函数返回值法..... 87
1.7 TC 集成环境的文本编辑命令.....13	4.2 一维数组元素排序函数..... 88
1.8 临时进入 DOS 状态.....14	4.2.1 选择法..... 88
第 2 章 用 Visual C++ 6.0 开发 C 程序.....15	4.2.2 冒泡法..... 89
2.1 Visual C++ 6.0 集成开发环境简介.....15	4.2.3 插入法..... 90
2.2 控制台应用程序的开发步骤.....16	4.2.4 希尔法..... 91
2.3 在 VC++6.0 开发环境中导入 已有的 C 程序.....19	4.3 创建年历文件..... 92
2.4 在 VC++6.0 开发环境下调试 C 程序.....21	4.4 Josephus 问题..... 101
2.5 程序的进一步调试.....22	4.5 求数学函数积分的通用模块..... 104
第 3 章 主教材习题参考解答.....28	4.5.1 矩形法..... 105
练习一.....28	4.5.2 梯形法..... 107
练习二.....29	4.5.3 辛普生法..... 109
练习三.....31	4.6 项目文件的处理..... 111
练习四.....33	第 5 章 二级等级考试部分题型求解.....114
练习五.....36	5.1 十进制与非十进制整数的转换.....114
练习六.....42	5.2 整数的分拆和重组.....119
练习七.....51	5.3 求最大数和最小数..... 123
练习八.....61	5.4 单词统计..... 125
练习九.....71	5.5 数据的重组与合并..... 127
	参考文献..... 132

第 1 章

Turbo C 2.0 集成开发环境介绍

1.1 概 述

进入 Turbo C 2.0 集成开发环境（以下简称“TC 开发环境”）后，屏幕如图 1.1 所示。

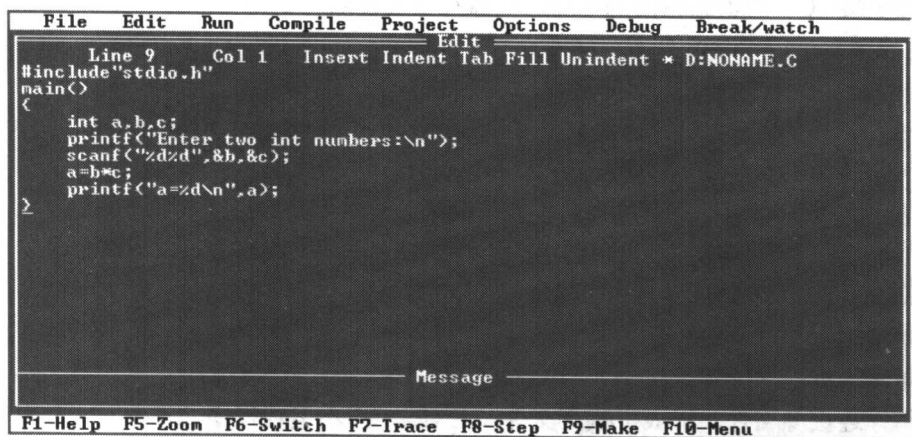


图 1.1 Turbo C 2.0 集成开发环境主界面

1. 主菜单和窗口

TC 屏幕最上面一行显示集成开发环境的主菜单：File、Edit、Run、Compile、Project、Options、Debug 和 Break/watch。

按 F10 键可以进入或退出菜单，用左、右方向键（←或→键）移动高亮度光标，选中后按回车键出现下拉菜单，再用上、下方向键（↑或↓键）选择下拉菜单中的命令。按 Alt 键加主菜单命令的首字符（F、E、R、C、P、O、D 或 B）键也可直接选中菜单命令。

主菜单下面有两个窗口，一个是编辑（Edit）窗口，另一个是信息（Message）窗口。

编辑窗口用来编辑源程序文件或者输入数据文件，该窗口的操作与一般字处理系统相似；信息窗口显示各种调试信息（如语法错误等）。

按功能键 F5，编辑窗口最大化，信息窗口消失；再按一下 F5 键，窗口又出现。

按功能键 F6 可以在编辑窗口和信息窗口之间切换光标位置。

2. 状态行

编辑窗口的第一行是状态行：

Line n Col m Insert Indent Tab Fill Unindent D:NONAME.C

其中:

Line n : 光标所在的行数 (第 n 行)。

Col m : 光标所在的列数 (第 m 列)。

Insert: 编辑窗口处在插入模式, 用 Ins 键 (或 Ctrl+V 键) 切换插入和改写模式。插入模式下, 输入的字符出现在光标处, 原光标处及其后的字符顺序后移; 改写模式下, 输入的字符替换光标处的字符, 光标后面的字符不变。

Indent: 自动缩进格式, 用 Ctrl+OI 键切换。

Tab: 建立制表状态, 用 Ctrl+OT 键切换。

Fill: 当建立制表状态 Tab 时, 编辑器将用制表符和空格符优化每一行的开始。

Unindent: 当光标在一行中第一个非空字符上或在空行上, 按退格键回退一级。用 Ctrl+OU 键切换。

D:NONAME.C: 当前编辑的源程序所在盘符及文件名。

3. 功能键提示行

TC 屏幕底端显示的是功能键提示行:

F1 - Help F5 - Zoom F6 - Switch F7 - Trace F8 - Step F9 - Make F10 - Menu

功能键作用如下:

F1: 激活帮助窗口, 提供有关当前位置的帮助信息。

F5: 放大/缩小当前窗口。

F6: 切换当前窗口。

F7: 单步运行程序, 跟踪进入函数内部。

F8: 单步运行程序, 不跟踪进入函数内部。

F9: 编译连接源文件, 形成可执行文件 (.EXE), 但不运行程序。

F10: 选择菜单。

1.2 使用 TC 的主要步骤

使用 TC 开发环境调试 C 程序, 主要包括 6 个步骤。

1. 进入 TC 环境

要进入 TC 开发环境, 应该运行 TC.EXE 文件。通常 TC.EXE 存放在 C:\TC 或 D:\TC 子目录下。假定 TC.EXE 存放在 D:\TC 子目录下, 要在 Windows 下运行 TC.EXE 文件可以双击该文件, 在 DOS 下运行该文件可输入如下行命令:

D:\TC\TC / (本书中 “/” 代表按回车键)

屏幕显示如图 1.1 所示的 TC 集成开发环境。

2. 输入程序

如果编辑窗口内已有程序, 可按 F10 键, 使菜单 File 高亮显示, 按回车键, 在弹出的菜单中选择 New, 再按回车键, 则原先在编辑窗口中的程序被清除了, 这时就可以开始输入新程序。为叙述简单, 以后把这一操作过程简单叙述为选择菜单 “File” / “New”。

操作练习 1:

(1) 选择菜单“File”/“New”，编辑窗口状态行显示最后一项为“NONAME.C”。

(2) 输入主教材第 1 章例 1.1 求两数之积的程序（去掉第 1 行注释），源程序清单如图 1.1 所示。

(3) 按功能键 F5 放大或缩小当前编辑窗口。

(4) 按功能键 F6 切换当前窗口。

3. 编译、连接和运行程序

仔细核对键入的程序，确认无误后，选择菜单“Run”/“Run”。如果程序正确，编译和连接没有问题，程序将正常运行并自动转入用户屏幕（User Screen），其上显示如下提示信息，要求用户输入两个整数（设为 5 和 8）：

```
Enter two int numbers:      (显示提示信息)
5 8✓                      (用户输入两个整数，用空格隔开)
```

输入两数后屏幕又返回到图 1.1 所示的编辑状态。注意在菜单命令“Run”/“Run”旁边的“Ctrl+F9”，说明运行程序也可以按快捷键 Ctrl+F9（同时按下 Ctrl 和 F9 键，然后松开）。每按一次 Ctrl+F9 键，程序就运行一次。

选择菜单“Run”/“User Screen”或按快捷键 Alt+F5，切换到用户屏幕，可以看到程序的运行状态和最后运行结果：

```
a = 40
```

按任意键，又切换到程序所在的编辑窗口。

注意，选择菜单“Run”/“Run”或按快捷键 Ctrl+F9，系统将已经修改过的程序实现编译、连接和运行，若程序没有语法错误，一般都可得到一个运行结果。若编译或连接过程中有错，则系统在信息窗口指出错误，此时高亮条停留在信息窗口的第一条错误信息上，按回车键光标回到编辑窗口并找到需修改的位置，等待对程序进行修改（也可以按 Alt+E 键直接进入编辑状态）。

一个程序经编译、连接和运行后，生成一个可执行文件。如果要多次运行程序，可多次按快捷键 Ctrl+F9 或多次选择菜单“Run”/“Run”，此时系统直接执行已生成的.EXE 文件。

操作练习 2:

① 在编辑状态下，将光标移到第一行，按 Home 键后（光标移到第一列）再按回车键，此时第一行下移，在空出的第一行上输入如下注释：

```
/* ex1_1.c the value of b*c */
```

说明程序的功能和准备使用的源程序文件名（当然用户可以使用别的注释说明内容）。

② 选择菜单“Run”/“Run”，输入两个整数（如-28 和 158）并观察输出结果。

③ 按 Ctrl+F9 键，输入两个整数（如 560 和-90）观察并分析输出结果。

4. 保存文件

程序运行完毕，应该保存源程序文件。选择菜单“File”/“Save”或按功能键 F2，如果是第一次存盘，系统会出现如下的提示框：

Rename NONAME
D:\TC\NONAME.C_

输入路径和文件名，TC 根据此信息存储源程序。注意：一般不用 NONAME.C 作为文件名存盘，这是开发环境提供的默认文件名。这里按程序第一行上的注释，输：

d:\ex1_1.c

按回车键后系统就按用户提供的文件名存盘，此时编辑窗口上方状态行上的文件名也已改变(它表示以 ex1_1.c 命名的文件保存在 D 盘的根目录中)，如图 1.2 所示。

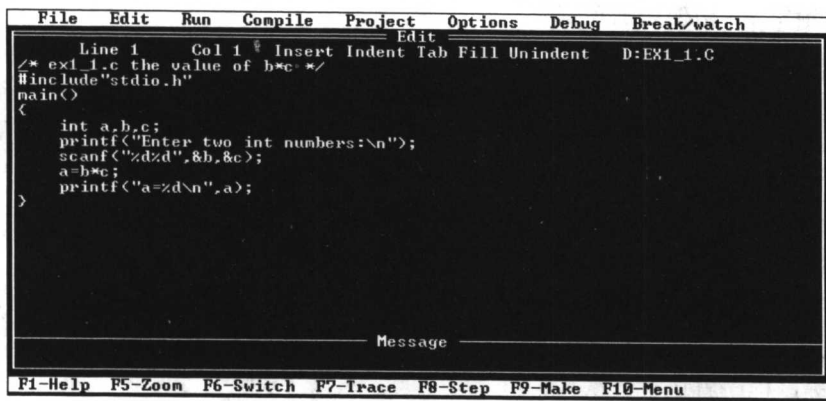


图 1.2 存盘后的编辑窗口

由图可见，编辑窗口状态行的最后一个单词已成为 D:EX1_1.C。

如果源程序已经按用户要求存过盘，但经修改后又要存储到其他磁盘中，或要将源程序改名存盘，可以选择菜单“File”/“Write to”，出现如下提示框：

NEW NAME

在提示框内输入新的盘符和文件名，则源程序又按新指定的盘符和文件名存盘。如果源程序修改后按原来的文件名存盘，只要按 F2 键或选择菜单“File”/“Save”即可。

操作练习 3:

- ① 选择菜单“File”/“New”后输入并编辑下列程序：

```
#include "stdio.h"
main()
{
    printf("The new file!\n");
}
```

检查无误后，按 Ctrl+F9 键（编译、连接和运行），再按 Alt+F5 键（观察运行结果），按任意键返回编辑窗口，并按功能键 F2 存盘（在出现的对话框内输入文件保存路径和文件名后按回车键）。

- ② 选择菜单“File”/“Write to”，在对话框中输入新盘符和文件名存盘。

5. 退出 TC 开发环境

选择菜单“File”/“Exit”或按快捷键 Alt+X 键，则可退出 TC 集成开发环境，回到原先的操作系统界面。假如在退出 TC 开发环境之前曾经编辑过源程序而又没有存盘，这时系统会出现如下提示框：

EX1_1.C not Saved? (Y/N)

提示是否要存盘。按 Y 键存盘后退出并返回操作系统；按 N 键不存盘退出。

操作练习 4:

选择菜单“File”/“Exit”或按快捷键 Alt+X 退出 TC 开发环境。退出 TC 开发环境后，检查一下文件是否已保存到指定路径。如果没有找到，请重做该部分的实验。

6. 导入程序

导入程序就是把已在磁盘中的源程序文件导入 TC 环境重新进行编辑或调试。进入 TC 开发环境后，选择菜单“File”/“Load”或按功能键 F3，在出现的对话框内输入盘符和路径后按回车键，在出现的文件框内（只显示扩展名为.C 的文件）通过方向键移动光标寻找所需的源程序文件，将光标定位在找到的源程序文件名上，按回车键把该文件导入 TC 环境。

导入文件也可以在进入 TC 开发环境时同时完成，这时只要在命令行上运行 TC.EXE 程序时，同时输入程序名，如

```
D:\TC\TC.EXE EX1_1.C✓
```

操作练习 5:

进入 TC 开发环境，通过“File”/“Load”导入源程序文件 EX1_1.C 或其他已存盘的源程序文件，重新修改，并编译、连接和运行。

1.3 修正编译过程中的语法错误

在编辑窗口中编写程序时，常常会出现一些不符合 C 语言规定的语法错误，使程序无法正常运行。当 TC 开发环境找到语法错误后，信息窗口中会显示出错信息。通常出错信息指出错误所在的行号及错误的性质，以方便程序员尽快找到源程序中的错误。如希望进一步了解出错的原因，可按 F1 键调用帮助文件。下面用实例说明语法出错及其修改的方法。

- ① 进入 TC 开发环境，导入程序 EX1_1.C。
- ② 在语句“int a,b,c;”中删除变量 c，语句变为：“int a,b;”。
- ③ 选择菜单 Run/Run 或按快捷键 Ctrl+F9，编译连接运行程序，屏幕上出现编译出错信息框：

Compiling		
Main file:	\QI\EX1_1.C	
Compiling:	EDITOR-->EX1_1.C	
	Total	File
Lines compiled:	219	219
Warning:	0	0
Error:	1	1
Available memory	217K	
ERRORS:	Press any key	

说明有一个编译错误，按任意键，切换到消息窗口中，出错信息为：

```
Compiling D:\QI\EX1_1.C
```

```
ERROR D:\QI\EX1_1.C 7: Undefined Symbol 'c' in function main
```

出错信息指出，源程序第7行有语法错误，并指出出错原因是在主函数 `main()` 中没有定义变量 `c`，而在第7行却使用了变量 `c`，这样就很容易找到错误了。按回车键（或按 F6 键）后，源程序对应错误的行高亮显示，按 Alt+E 键后高亮消失，进行纠错编辑。

下面列出部分常见的编译错误：

Statement missing ;	语句后缺少 “;” 号
Possible use of x before definition	在定义之前就使用 x
Redeclaration of x	重复定义了 x
Out of memory	内存不够用
Wrong number of arguments	调用函数时参数数目错
x declared but never used	说明了 x，但没有使用
Type mismatch in parameter x	参数 x 类型不匹配
Undefined symbol x	没有定义的符号 x
Illegal use of pointer	非法使用指针
Division by zero	除数为零
Declaration syntax error	说明中出现语法错误
Argument list syntax error	参数表语法错误
Ambiguous symbol x	不明确的符号 x
Call to function with no prototype	调用函数时没有函数原型的说明

④ 修改程序。按 F6 键，从消息窗口切换到编辑窗口，高亮条停留在语法出错行上。通常语法出错在该行或该行之前（附近）。增加定义变量 `c` 后，程序恢复正常运行。注意，有时语法错误不一定是在高亮条所停留的行上，而是在它的上一行。例如上一行少了一个分号，则系统将这一行与下一行作为一行对待，因此光标就可能停留在下一行。

操作练习 6：

在程序 EX1_1.C 中，删除语句 “`int a,b,c;`” 中最后的一个分号，写出编译源程序时的出错信息，并指出这些信息的含义是什么。

1.4 修正程序的算法错误

1. 程序的单步运行

若程序能一步一步地执行，就能观察到源程序的执行流程是否与原先设计的流程相一致。如果源程序的执行流程与设计中的不符，一般就能寻找到引起错误的原因了，进而可以改进算法来纠正错误。

选择菜单 “Run” / “Step over”，或者按功能键 F8，程序将单步运行。这时一条高亮条出现在 `main()` 行上。如果再按一次 F8，高亮条移到 `main()` 函数中的第一个可执行语句行上；再按一次 F8 键，高亮条移到 `main()` 函数中的第二个可执行语句行上，……不断按 F8 键，可以看到高亮条不断往下移动，每按一次执行一条可执行语句，直到程序执行完毕。由此可看清楚程序

的执行流程。

也可以选择菜单“Run”/“Trace into”，或者按功能键 F7，同样可以进行单步运行。两者的区别在于 F8 把函数作为一条语句来执行，而 F7 则跟踪进入函数内部的每一条可执行语句。

操作练习 7:

(1) 含判断语句程序的单步运行

输入以下求两数中最大值的程序:

```
main()
{
    int a,b,max;
    printf("Input two integers: ");
    scanf("%d %d",&a,&b);
    if (a>b) max=a;
    else max=b;
    printf("max=%d\n",max);
}
```

单步运行上述程序两次。第一次 a、b 取值分别为 -36 和 58，第二次 a、b 取值分别为 88 和 25，看一下单步运行程序时，两次所执行的语句次序是否一样。

再输入以下求两数之和及之差的程序:

```
main()
{
    int a,b,c;
    printf("Input two integers: ");
    scanf("%d %d",&a,&b);
    c=a+b ;
    printf("a+b=%d\n",c);
    c=a-b;
    printf("a-b=%d\n",c);
}
```

试输入 100 和 2300，查看运行结果；再输入 23000 和 20000，查看运行结果，并分析结果的正确性，若有错分析原因。

(2) 含循环语句程序的单步运行

输入如下求 $s=100+110+120+\cdots+190$ 的程序:

```
main()
{
    int s=0,i=100;
    while(i<200)
    {
        s=s+i;
        i=i+10;
    }
    printf("s=%d\n",s);
}
```

单步运行上述程序，看一下在循环体中单步运行的次序是怎样的，并解释这样运行次序的原因。

如果把 `while(i<200)` 误写成 `while(i>200)`，再单步运行程序，看看这时单步运行的次序，并解释这时的运行结果为什么是 `s=0`。

(3) 含函数程序的单步运行

输入如下程序：

```
main()
{
    int i,j,k;
    int gcd(int,int);
    printf("input two integers:");
    scanf("%d%d",&i,&j);
    k=gcd(i,j);
    printf("greatest common divisor is %d\n",k);
}

int gcd(int a,int b)
{
    int temp;
    while(b)
    {
        temp=a%b;
        a=b;
        b=temp;
    }
    return a;
}
```

分别用功能键 F8 和 F7 来单步运行该程序，看一看两者单步运行有何差别。

2. 变量值的观察

在单步运行程序时，若能同时观察到变量的变化情况，就能方便地分析程序执行过程是否正确。要观察变量值，可选择菜单“Break”/“Watch”/“Add watch”或按快捷键 Ctrl+F7，这时屏幕上会出现一个 Add Watch 输入框，在框中输入要观察的变量名，然后按回车键，切换到监视 (Watch) 窗口，出现被观察变量的变化值。

如果要观察多个变量值，可以多次按 Ctrl+F7 键，每次增加一个观察变量。

如果要从监视窗口中删除所有的观察变量，可选择菜单“Break”/“Watch”/“Remove all watches”。如果只希望删除某个观察变量，按功能键 F6，切换到监视窗口中，选中待删除的观察变量，然后选择菜单“Break”/“Watch”/“Delete watch”。

操作练习 8：

(1) 导入或输入如下程序：

```
main()
{
```

```

int a,b,max;
printf("Input two integers: ");
scanf("%d %d",&a,&b);
if (a>b) max=a;
else max=b;
printf("max=%d\n",max);
}

```

按 Ctrl+F7 键, 增加观察变量 a、b 和 max (注意观察三个变量要分别按三次 Ctrl+F7 键), 单步运行程序, 观察 a、b、max 值的变化情况, 不同的输入得到不同的结果。

把源程序中语句 “scanf (“%d%d”,&a,&b);” 改成 “scanf (“%d%d”,a,b);”, 然后运行程序 (输入语句中的变量要求用地址信息, 改动以后出现 “语法” 错误, 但这个错误编译系统检查不出来, 可能是搞不清楚变量定义时的那个不确定的值是否是地址值), 观察变量 a、b 及 max 的变化情况。在此情况下, 输入值发生了变化时, 变量 a、b、max 的值却没有变化并明显是错误的。此时可选择 “Compile” / “Compile to OBJ” 命令编译程序, 此时出现如下编译对话框:

Compiling		
Main file:	\QI\EX1_1.C	
Compiling:	EDITOR-->EX1_1.C	
	Total	File
Lines compiled:	219	219
Warning:	4	4
Error:	0	0
Available memory	287K	
Warnings:	Press any key	

没有错误但有警告, 按任意键后回到调试状态, 此时在信息窗口看到如下警告信息:

Compiling D:\QI\EX1_1.C

Warning: D:\QI\EX1_1.C 7: possible use of 'c' before definition in function main

Warning: D:\QI\EX1_1.C 7: possible use of 'b' before definition in function main

这个警告没有明确指出错误的性质, 并且程序编译后生成 OBJ 文件, 通过连接仍可以产生可运行 EXE 文件。在这种情况下, 应对警告信息有足够的重视, 寻找警告原因, 排除错误, 纠正运行错误。

(2) 导入或输入如下程序:

```

main()
{
    int s=0,i=100;
    while(i<200)
    {
        s=s+i;
        i=i+10;
    }
    printf("s=%d\n",s);
}

```

```

}

```

按 Ctrl+F7 键, 增加观察变量 s 和 i。单步执行程序, 观察变量 s 和 i 的变化情况。

(3) 导入或输入如下程序:

```

int sub(int c,int d)
{
    c=c*c;
    d=d*d;
    printf("c=%d\td=%d\n",c,d);
    return c-d;
}

main()
{
    int i=6,j=8,k;
    int sub(int,int);
    k=sub(i,j);
    printf("k=%d\n",k);
    printf("i=%d\tj=%d\n",i,j);
}

```

按 Ctrl+F7 键, 增加观察主函数中的变量 i、j、k 和函数 sub() 中的变量 c、d, 按 F7 键单步运行观察这些变量值的变化。

3. 程序的多步运行

当程序较长或程序中有较多的循环时, 单步运行将耗时甚长, 降低了程序调试的效率。这时可采用多步运行的调试方法。

(1) 程序运行到光标处

把光标移动到指定语句上, 然后按功能键 F4 或选择菜单 “Run” / “Go to cursor”, 该行高亮显示, 此时程序已经从 main() 行开始运行到高亮条所在行了。要程序继续运行下去可以按 Ctrl+F9。若要取消由于按功能键 F4 或选择菜单 “Run” / “Go to cursor” 设定的高亮条, 可以将光标移动到该行, 再按一次 F4 键, 高亮条即消失。

操作练习 9:

导入如下程序:

```

int sub(int c,int d)
{
    c=c*c;
    d=d*d;
    printf("c=%d\td=%d\n",c,d);
    return c-d;
}

main()
{
    int i=6,j=8,k;
    int sub(int,int);
    k=sub(i,j);
}

```

```
printf("k=%d\n",k);
printf("i=%d\tj=%d\n",i,j);
}
```

按 Ctrl+F7 键观察变量 i、j、c、d，然后将光标移到“k=sub(i,j)”行上，按 F4 键，该行显示高亮条，此时程序已经运行到该行，这可以从监视窗口中变量值的显示得到验证：

```
d: Undefined symbol 'd'
c: Undefined symbol 'c'
j: 8
i: 6
```

main()函数中的两个被观察变量 i 和 j 已经有值；sub()函数中的 c 和 d 分别显示 Undefined symbol 'c' 和 Undefined symbol 'd'。

再将光标移到函数 sub()中的“return c-d;”行上，按 F4 键，高亮条转移到该行上，此时程序已经运行到该行，监视窗口中变量值的显示是：

```
d: 64
c: 36
j: Undefined symbol 'j'
i: Undefined symbol 'i'
```

可见，监视窗口只显示当前正在运行函数中被观察变量的值，此时其他函数中的被观察变量被认为是 Undefined symbol（未定义的符号）。

从高亮条行要继续运行下去可按 Ctrl+F9 键，程序运行结束。要撤消行上的高亮条只要移动光标到该行上再按一次 F4，高亮条消失。

(2) 程序运行到断点处

断点是一种设定标志，在程序运行过程中（按 Ctrl+F9 键），遇到断点，程序就停止执行，如果还要继续运行程序（可以再按一下 Ctrl+F9 键），程序从断点处继续开始运行，直到下一个断点或到程序运行结束。

断点设置的方法是把光标移动到欲设置断点的语句处，按快捷键 Ctrl+F8 或选择菜单“Break”/“Watch”/“Toggle breakpoint”，这时所选语句处出现一个红色光条，说明断点已经建立。在一个源程序中根据需要可以设置多个断点。

撤消断点的方法：选择菜单“Break”/“Watch”/“Clear all breaks”，则撤消所有已设定的断点。如果只要撤消某一个断点，把光标移动到待撤消的断点处，按快捷键 Ctrl+F8 或选择菜单“Break”/“Watch”/“Toggle breakpoint”，就可以撤消原先的断点（建立和撤消一个断点的操作相同）。

操作练习 10:

输入如下用“选择法”对一维数组元素排序的程序。

```
#include "stdio.h"
#define N 10
main()
{
    int i,j,k,t,a[N]={90,87,12,-22,55,-77,120,33,8,44};
    for(i=0; i<N-1; i++)
```