

C 语言程序设计上机指导与练习

钟廷志 赵建刚 编著

人 民 邮 电 出 版 社

图书在版编目(CIP)数据

C 语言程序设计上机指导与练习/钟廷志, 赵建刚编著. —北京: 人民邮电出版社, 2004.6
(中等职业学校计算机系列教材)

ISBN 7-115-12154-0

I. C... II. ①钟...②赵... III. C 语言—程序设计—专业学校—教学参考资料 IV. TP312
中国版本图书馆 CIP 数据核字 (2004) 第 048545 号

内容提要

本书是《C 语言程序设计》的配套教材, 内容以实验操作为主, 重点培养学生的实际动手编程能力。全书共分 10 个实验, 包括运行环境、C 语言基本知识、C 语言程序结构、函数、数组、指针和预处理、共用体等部分的上机操作实验。每个实验包括实验目的、预习知识、注意事项、错误改正、上机实例、程序调试和案例模仿等部分, 通过实验指导, 学生能够事半功倍地掌握相关的知识点和操作方法。

本书适合作中等职业学校“C 语言程序设计”课程的上机教材, 也可作为 C 语言程序设计初学者的自学参考书。

前 言

本书是《C 语言程序设计》的配套教材，以实验操作为主，通过有针对性的上机操作，使学生掌握 Turbo C 的基本操作方法和编程应用技巧。

教师一般可用 40 个课时来讲解《C 语言程序设计》的内容，然后配合本上机指导，再分配 40 个课时作为上机时间，即可顺利完成教学任务。总共需要约 80 个课时。两本教材中的内容可以满足扩充至 120 课时所需的内容要求，教师在实际授课过程中也可以根据需要对学时进行适当的调整。

本书结构与《C 语言程序设计》一书的结构相对应，每个实验基本上由以下几个主要部分组成：

- 实验目的：罗列出本实验要达到的目的，教师可用它作为简单的备课提纲，学生可通过实验目的对本实验的内容有一个大体的认识。
- 预习知识：给出本实验的所要预习的知识点。
- 注意事项：给出本实验中容易出现错误的地方，加深读者对相关知识的正确理解。
- 改正下列错误：列出了本章中的常见错误，并进行改正。使读者能事半功倍地掌握相关的函数、语句、命令等的正确使用。
- 上机实例：运用相关知识进行典型程序的设计。
- 程序调试：通过对有误程序的调试，进行一步掌握本章的内容。
- 案例模仿：通过对功能类似程序编写的模仿，全面、扎实地掌握本章知识点。

本书是专门为中等职业学校编写的，适合作 C 语言编程上机教材，也可作初学者自学 C 语言程序设计的参考书。

由于作者水平有限，疏漏之处敬请各位老师和同学指正。

作者

2004 年 4 月

目 录

第 1 章	练习使用 C 程序的开发环境.....	1
第 2 章	结构、变量、运算符、表达式	11
第 3 章	输入输出函数和实现程序的步骤	24
第 4 章	顺序结构程序	33
第 5 章	选择结构程序设计	42
第 6 章	循环控制程序设计	55
第 7 章	函数的使用	68
第 8 章	数组	81
第 9 章	指针	95
第 10 章	编译预处理、结构体、共用体	106
附录	常用资料	116
F.1	Turbo C 编译出错信息	116
F.2	常见函数、语句和命令的规范	125

第1章 练习使用 C 程序的开发环境

一、实验目的

- 熟悉 Turbo C 的运行环境。
- 了解 UNIX、DOS 下 C 的运行环境。

二、3 种运行环境的介绍

- 用 Turbo C 运行 C 程序。
- 在 UNIX 操作系统下运行 C 程序。
- 在 DOS 下运行 C 程序。

三、Turbo C 运行环境简介

C 语言是功能强大的程序设计语言，它不仅具有一般低级语言的许多特点，而且还具有一些其他高级语言的特点，因此特别适合于编写系统软件（例如，著名的 UNIX 操作系统就是用 C 语言来编写的），是目前应用最广泛的程序语言之一。

Turbo C 是 Borland 公司开发的 C 程序开发平台，本书它具有良好的用户界面和丰富的库函数，功能十分强大，可以完成对 C 语言程序的编辑、编译、链接、调试以及程序运行等工作。与其他开发平台相比，Turbo C 软件小、安装方便、界面友好、速度快、效率高、功能完善。本书使用 Turbo C 2.0 进行 C 程序开发。

(1) Turbo C 的安装

用户在进行 C 程序的开发之前，需要把 Turbo C 安装在硬盘上。安装时，按安装程序的提示进行即可。安装程序名为“install.exe”，安装过程很简单，运行“install.exe”文件，按照系统提示作一些必要的参数设置后，安装程序就会自动在硬盘上建立一个 C 程序开发系统。在安装时，可以根据需要选择存放文件的目录，如无特殊要求，一般选择默认的方案。

Turbo C 对系统配置要求不高，只需要很小的系统开销，它仅占用 384KB 的内存，其系统文件占用的磁盘空间不到 3MB，一般要求计算机有两个驱动器（1 个软盘驱动器和 1 个硬盘驱动器），显示器可以是单色的，也可以是彩色的，可以不用配置鼠标。

(2) 启动 Turbo C

要编辑并运行 1 个 C 程序，进入 Turbo C 系统有以下方法。

由 DOS 平台进入 Turbo C:

假如 Turbo C 安装在 D 盘，使用下面的 DOS 命令。

```
D: \>CD\TC<回车>
```

```
D: \TC>tc<回车>
```

这样就能进入 Turbo C 环境了，屏幕将显示 Turbo C 的工作窗口。

由 Windows 平台进入 Turbo C:

在这种状态下，进入 Turbo C 系统的方法主要有以下几种。

- 依次选择【开始】/【程序】/【MS-DOS 方式】选项，进入【DOS 方式】窗口，在该窗口中使用上述 DOS 命令，进入 Turbo C。
- 通过【资源管理器】窗口找到文件夹“TC”中的“tc.exe”文件，双击该文件



名，即可进入 Turbo C。

- 通过【资源管理器】窗口找到文件夹“TC”，为其中的“tc.exe”文件创建快捷方式，然后把它发送到桌面上。此时，在桌面上将出现一个“TC.EXE”的图标，双击该图标即可进入 Turbo C。

(3) Turbo C 的工作窗口

正确启动 Turbo C 后，就会在屏幕上出现 Turbo C 的工作窗口，如图 1-1 所示。这个工作窗口由 4 部分组成，分别介绍如下。

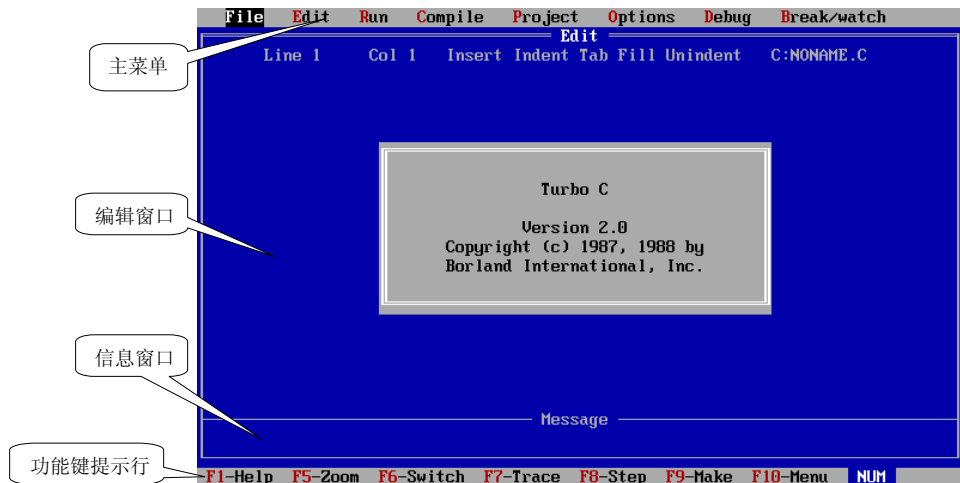


图1-1 Turbo C 的工作窗口

主菜单：

主菜单，位于窗口中最上面一行，包括【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）等 8 个下拉菜单选项，分别用来实现各项操作。

【编辑】窗口：

位于主菜单下部，程序文件的编辑工作就在该窗口中进行。在编辑框的上面一行显示各种编辑状态，其含义如下。

- **【Line】**：当前光标所在的行号。
- **【Col】**：当前光标所在的列号。
- **【Insert】**：编辑器处于插入或替换模式，按 **Insert** 键在插入和替换两种状态下切换。当为插入模式时，在当前可以输入要对程序补充的内容；当为替换模式时，输入的内容将替换后面的字符。
- **【Indent】**：自动缩进设置。
- **【Tab】**：设置是否能插入 **Tab** 键。
- **【C:NONAME.C】**：表示当前正在编辑文件的文件名和路径。

在本书的叙述中，命令 **Ctrl+QT**、**Ctrl+KD**、**Ctrl+KV** 等表示按下 **Ctrl** 键后不要放开，再键入后面的字母（如 QT），这时在【编辑】窗口左上角会出现相应的（如“^KT”）标志。

【信息】窗口：

在 Message 栏下面，用于显示编译和链接时产生的信息，它在编辑源程序时不起作用，只



有在编译链接时显示程序的信息。例如此处将显示程序编辑过程中的各类出错信息，帮助用户完成对程序的调试工作。

功能键提示行：

Turbo C 将常用的功能处理为热键，便于用户实现快速操作。常用的热键有以下几种。

- F1—Help (帮助)：使用过程中，如遇到问题，按下 **F1** 键会显示帮助信息。
- F5—Zoom (分区控制)：**F5** 键用于扩大【编辑】窗口，以便能容纳和显示较长的源程序和信息。当【编辑】窗口处于激活状态时，按下 **F5** 键，【信息】窗口消失，便于编辑和修改程序；当【信息】窗口处于激活状态时，按下 **F5** 键，则最大化显示【信息】窗口，【编辑】窗口消失，便于用户查看信息；若再按 1 次 **F5** 键，就会恢复原状。
- F6—Switch (转换)：用于实现【编辑】窗口和【信息】窗口之间的切换。按 **F6** 键就激活【信息】窗口，此时【编辑】窗口就不能工作。若再按 1 次 **F6** 键，就又激活【编辑】窗口，此时能在【编辑】窗口中编辑源程序。
- F7—Trace (跟踪)：用于跟踪程序的运行情况，检查程序出错的地方。
- F8—Step (按步执行)：按 1 次 **F8** 键执行 1 个语句。
- F9—Make (生成目标文件)：进行编译和链接，生成目标文件“.obj”和可执行文件，但不进行运行。
- F10—Menu (菜单)：回到主菜单，激活第 1 个主菜单：【File】。

以上对工作窗口只做了简单的说明，下面将介绍各项菜单的功能。

四、主菜单中各项菜单的功能说明

进入 Turbo C 的运行环境后，对主菜单的操作有两种方法：

- 按 **F10** 键把光标移动到主菜单行，然后按 **→** 和 **←** 键把光标移动到希望进入的菜单处，按回车键。
- 直接按 **Alt+x** (x 为某菜单项的第 1 个字母，例如按 **Alt+F** 就是选择了【File】菜单项，进入文件菜单)。

下面对 Turbo C 运行环境的几个主要菜单和重要命令进行讲解。

【File】菜单：

将光标移动到【File】菜单项上时，按回车键打开下拉菜单，如图 1-2 所示。

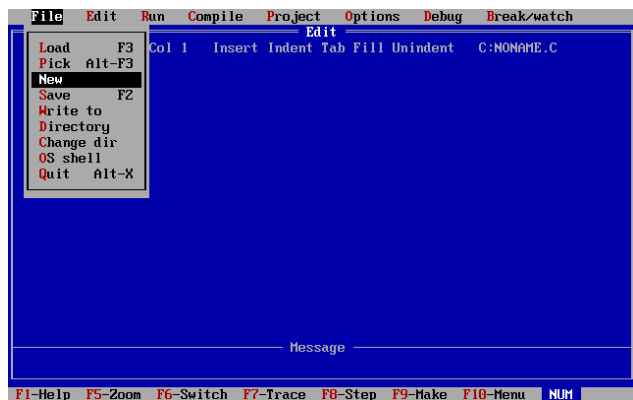


图1-2 【File】下拉主菜单



下面介绍几个主要菜单的功能:

(1) **【New】**(新建)

打开 1 个新的工作窗口, 当前编辑的文件为最新的文件, 文件缺省名为系统自动生成的文件名 “NONAME.C”。

(2) **【Open】**(打开)

导入 1 个文件到编辑器, 即载入 1 个新的程序, 可以用通配符 (*) 进行列表选择。一般 C 语言程序是后缀名为 c 的文件 “*.c”, 也可以直接键入文件名及其路径。用户可以将已经编好的程序通过此功能调到**【编辑】**窗口中进行修改、编译、链接和运行。这样可以方便快捷地打开最近操作过的文件, 可以用 **F3** 热键实现同样的操作。

(3) **【Save】**(保存)

将编辑器中的文件保存, 也可以用 **F2** 热键, 当第一次保存文件时, 系统提示输入要保存文件的名字, 默认文件名为 “NONAME”, 以后每次要保存编辑结果只需按 **F2** 即可。

(4) **【Os shell】**(暂时退出)

暂时退出 Turbo C, 转到 DOS 提示符下, 也可以用 EXIT 命令退回到 Turbo C。

(5) **【Quit】**(退出)

退出 Turbo C, 返回 DOS 提示符下或 Windows 系统。

【Edit】 菜单:

可以用 **Alt+E** 命令选择 **【Edit】** 菜单, 按回车键后进入编辑状态, 此时, 可对源文件进行编辑, 其编辑方法与一般的文字处理程序类似。

【Edit】 菜单没有下拉菜单项。编辑命令可分为光标移动、插入/删除、块操作和其他命令 4 种。如下所示为 Turbo C 编辑程序常用命令。

(1) 光标移动

- **←**: 左移一格。
- **→**: 右移一格。
- **↑**: 上移一格。
- **↓**: 下移一格。
- **Page Up**: 前移一格。
- **Page Dn**: 后移一格。
- **Home**: 移到行首。
- **End**: 移到行尾。

(2) 插入操作

- **Insert**: 插入操作打开或关闭。
- **Del**: 删除光标所指字符。
- **Backspace**: 删除光标前 1 个字符。
- **Ctrl+N**: 插入 1 行。
- **Ctrl+Y**: 删除光标所在行。

(3) 块操作

- **Ctrl+KB**: 块头标记。
- **Ctrl+KK**: 块尾标记。
- **Ctrl+KC**: 块复制。



- **Ctrl+KY**: 块删除。
 - **Ctrl+KV**: 块移动。
- (4) 其他命令
- **Ctrl+U**: 撤消正在进行的操作。
 - **Ctrl+KD**: 文件存盘退出。
 - **Ctrl+KQ**: 文件不存盘退出。

在【编辑】窗口的最上面一行出现“Insert”提示时，表示当前处于插入编辑状态，此时从键盘上输入的字符就插入在光标停留的地方。“Insert”提示符消失时，表示当前处于重写状态（覆盖状态），此时从键盘输入的字符将取代光标所在处的字符。按 **Insert** 键则可以改变“插入”或“重写”的设置。

Ctrl+F1 键的作用是显示在线帮助信息。用法是把光标移到任意关键字或函数名处按 **Ctrl** 键的同时按下 **F1** 键，此时，会在屏幕上显示出该关键字或函数的使用说明信息。为了使用在线帮助，必须把帮助文件“TCHELP.TCH”拷贝到当前目录 PATH 说明的目录中。

【Run】菜单：

【Run】菜单的作用是运行和调试程序，如图 1-3 所示，可用 **Alt+R** 打开【Run】菜单，有以下是一些常用命令。

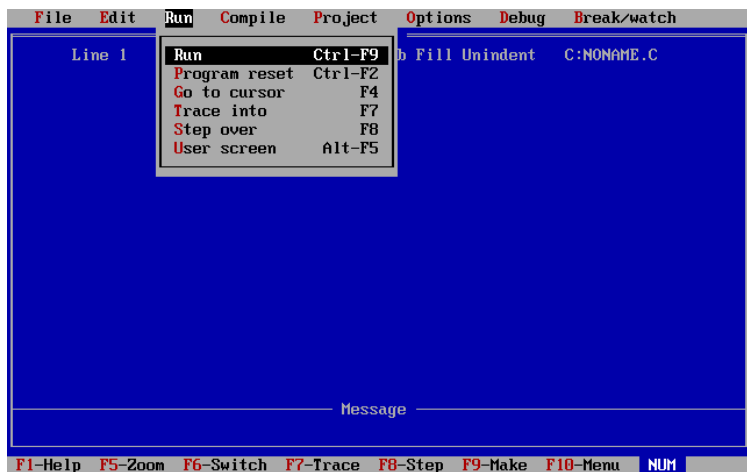


图1-3 【Run】下拉主菜单

(1) 【Run】（运行）

用该命令运行经过编译和链接后的可执行文件，以获得最后的结果。该命令对应的热键是 **Ctrl+F9**。

(2) 【Go to Cursor】（执行到光标所在行）

在调试程序时，使程序执行到【编辑】窗口中光标所在位置。

(3) 【Trace into】（跟踪进入）

在程序运行时对运行状态进行跟踪，它可以跟踪到函数内部以便用户观察程序的运行情况，例如查看变量当前的值。该命令对应的热键是 **Ctrl+F7**。

(4) 【Step over】（单步执行结束）

它表示在程序执行过程中，用户可根据此命令对程序进行分段执行。该命令对应的热键是 **Ctrl+F8**。



不管光标停留在哪个菜单，在 Turbo C 系统的最下端会出现提示信息，内容为当前菜单的主要功能，这样有利于用户对程序进行各种功能的操作。

例如：

F1 Help | Make and run the current program

Turbo C 用户窗口中除以上几个主要菜单外，还有以下菜单。

【Compile】菜单：

用于对界面进行动态编辑，菜单命令如图 1-4 所示，其中两个主要选项的功能如下。

- **【Compile】**：编译 C 程序。
- **【Link】**：链接 C 程序。

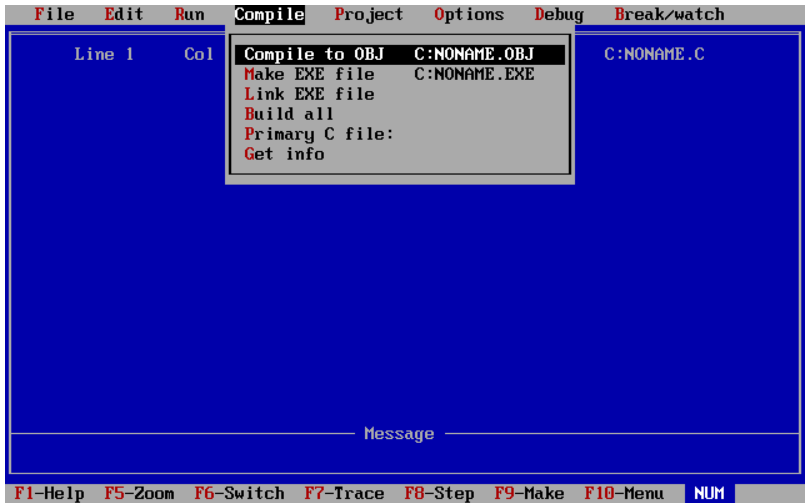


图1-4 【Compile】菜单

【Debug】菜单：

用于调试程序，可以打开一个观察窗口来检查程序中变量的值，如图 1-5 所示。其功能主要是检查变量当前的值、查看修改一个变量的值时其他变量值的变化情况等。使用这个菜单中的命令可以使调试程序更加方便，在调试复杂的程序时尤其重要，通过它可以节约修改程序的时间、很快地找出程序出错的地方。熟练使用这个菜单，可以大大提高工作效率。



图1-5 Debug 菜单



五、实例操作

编写 1 个程序求 $1+2+3+4+5$ 的值，并且运行出结果。

编写此程序主要有以下几个步骤。

1. 新建文件。

打开 Turbo C，建立 1 个新的程序文件，如图 1-6 所示，当光标停留在【File】菜单的【New】菜单选项上时敲回车键或者单击该菜单项即可。

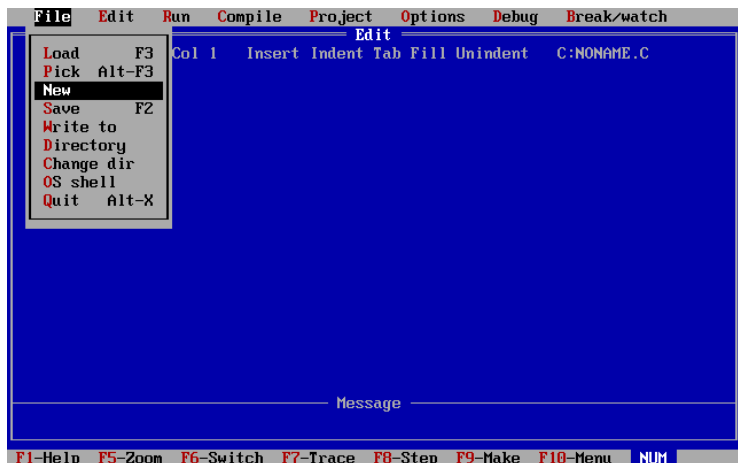


图1-6 新建程序文件

2. 编辑程序。

在【编辑】窗口中编辑程序，如图 1-7 所示。

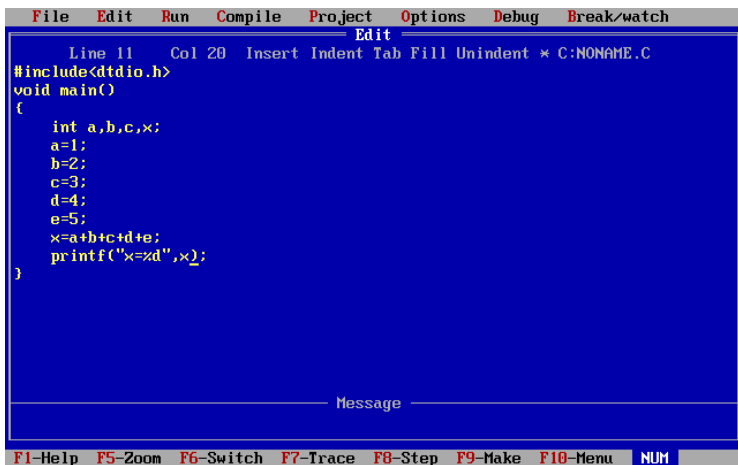


图1-7 在【编辑】窗口中编辑程序

3. 编译和链接程序。

对程序进行编译、链接，生成可执行文件。

(1) 编译程序。

在【Compile】下拉主菜单中选取【Compile】命令，编译程序。

(2) 链接程序。

如果编译过程中发现错误，需要根据系统提示修改错误。当编译顺利通过后，在



【Compile】下拉主菜单中选取【Link】命令，链接程序，生成可执行文件。

系统在编译过程中往往会发现程序中的错误，并在【信息】窗口中列出错误信息。例如，如果本例中程序设计不正确，则会在编译过程中出错，系统给出如 1-8 所示的出错信息。

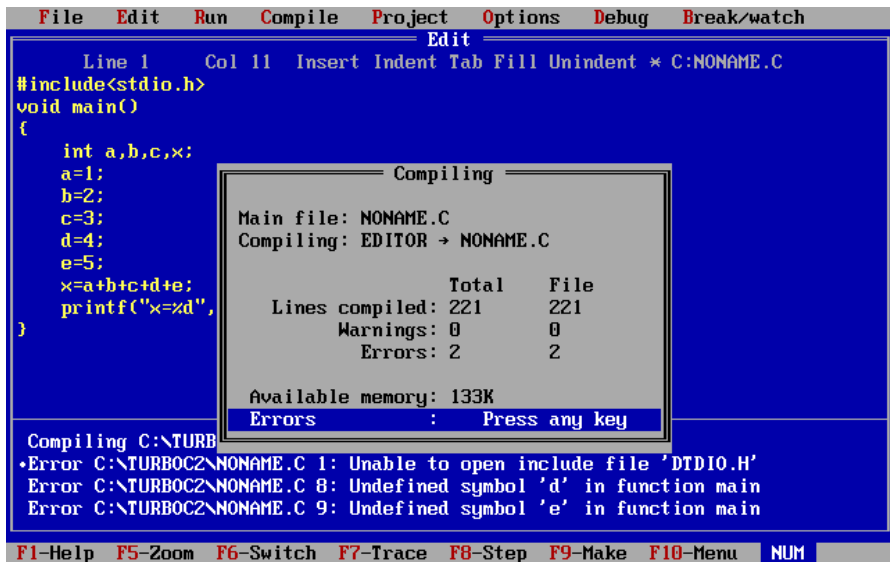


图1-8 编译程序时的窗口

4. 修改程序。

下面逐一修改图 1-8 所示的错误。

按一下回车键后，出现如图 1-9 所示界面：

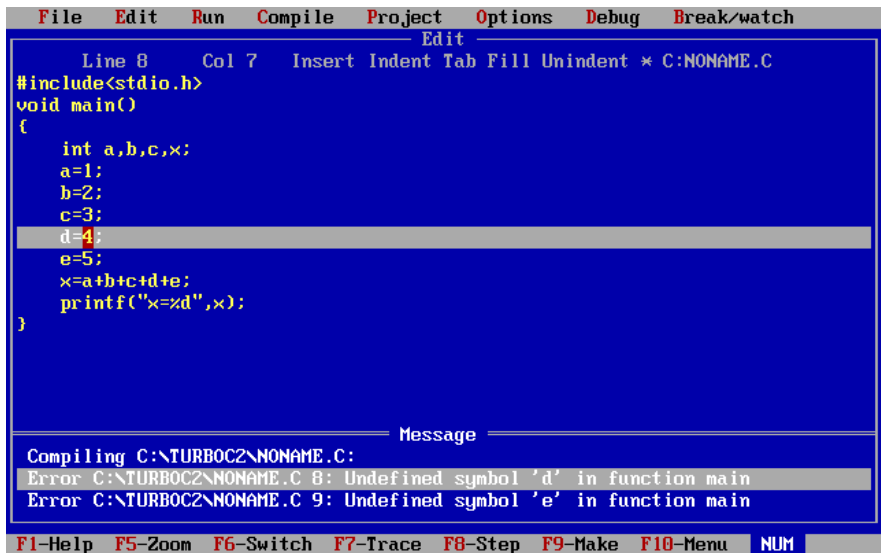


图1-9 出错信息界面（一）

在【Message】栏下，有两条出错信息，其中一条错误信息显示为灰色，即为当前激活的出错信息，其具体位置在编辑框中的红色标识处。出错原因为：变量 d 没有定义而在程序中又使用了该变量，这是因为 C 语言程序中要求所有的变量必须先定义后使用。

改正此处错误后，继续改正下一处错误，按一下向下箭头键并回车，出现如图 1-10 所示



界面:

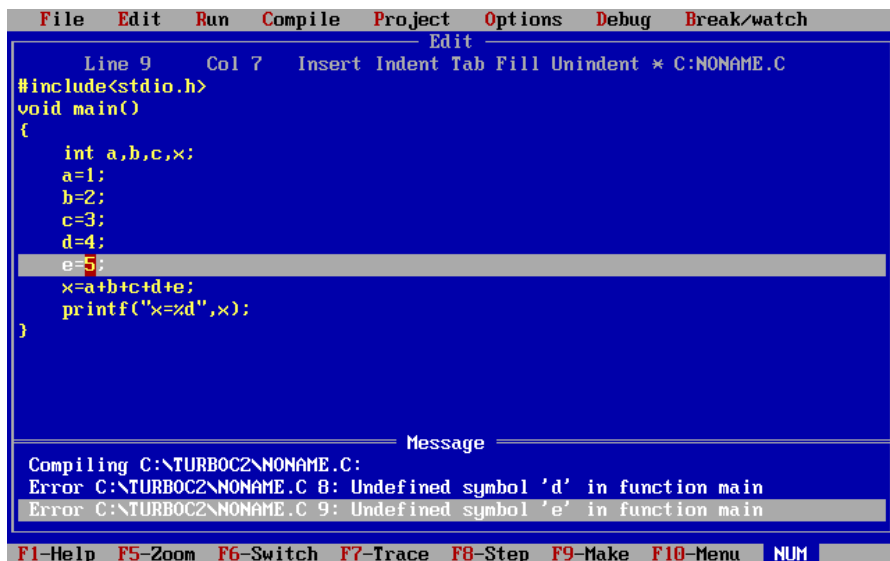


图1-10 出错信息界面（二）

这里提示变量 `e` 出现了同样的问题，根据提示进行相应的改正，即把程序中的“`int a,b,c,x;`”改为“`int a,b,c,d,e,x;`”。

在实际的程序调试中，往往出现这种情况，当前面的某些错误修改后，后边的错误就随之解决了。总之，错误信息框中的信息有利于用户对程序的调试，在实际的运用中，要重视和利用这些信息。

5. 运行。

经相应的改正后再进行编译、链接，系统无出错信息，说明已通过编译、链接，可以运行该程序，执行【Run】下拉主菜单中的【Run】命令或者按下 **Ctrl+F9** 运行程序。

6. 查看结果。

执行【Window】下拉主菜单中的【User screen】命令或按热键 **Alt+F5** 查看程序运行结果，本程序运行后的结果如图 1-9 所示。



图 1-11 程序运行的结果



由图 1-11 可知，该程序的运行结果为：

x=15

注意：该程序的运行结果只为“x=15”，图 1-11 所示界面上的其他部分为由 DOS 进入 Turbo C 2.0 的相关命令和其他信息。

7. 核实。

在程序通过编译、链接并运行出结果后，还要进一步核实结果是否正确，这主要是防止程序既没有语法错误也没有算法错误。C 语言是不能对算法进行检查的，所以核实运行结果，检查程序是否有逻辑上的错误既是重点也是难点。

六、 模仿练习

上机编辑、运行以下程序并核实其结果是否正确。

```
#include<stdio.h>

void main()
{
    int a,b,c;
    a=3;
    b=5;
    c=a+b;
    printf("%d",c);
}
```

第2章 结构、变量、运算符、表达式

一、实验目的

- 掌握 C 程序的完整结构。
- 学会对整型、字符型、实型变量的正确定义。
- 能对变量正确赋值。
- 掌握对运算符和表达式的正确使用。
- 熟悉对程序的调试过程。

二、预习知识

- C 程序的结构。
- 各种变量的含义。
- 赋值的相关知识。
- 各种运算的优先级和结合方式。
- 有关表达式的相关知识。

三、注意事项

(1) C 程序的结构。

C 程序的完整结构如下：

```
#include<头文件>
void main()
{
    定义部分
    其他语句
}
```

- “#include<头文件>”的作用是把该程序中所要用到的函数包含到该程序中，直接使用 C 语言中现成的函数，减少编写程序的难度，提高编写程序的效率。在 C 程序中，通常都要用到一个头文件，即“stdio.h”，因为一个程序中一般都要有数据输入和输出操作。由于“#include<头文件>”是“命令”而不是语句，所以其后一定不能有“;”，否则为语法错误。使用时有两种形式，下面以“stdio.h”为例进行说明。

形式 1:

```
#include<stdio.h>
```

形式 2:

```
#include"stdio.h"
```

- “void main()”是主函数的标志，每一个 C 程序都是从这里开始执行的，如果没有这一部分，程序就不知从何处开始执行，当然编译时也不能通过。不过，如果不要其中的“void”，程序也能编译通过并执行，但会出现警告信息。为了最大限度地避免出错，提倡使用“void”这一关键字。main 后的小括号不能省略，它是函数中必不可少的部分，否则会出现语法错误。可以把它理解为“函数”之



意。

- “定义部分”和“其他语句”构成函数体，其后必须加“;”。一个函数可以为空，即没有任何语句。这种函数没有任何语句可执行，相应地也不会有任何结果。
- 一个函数可以没有语句，但大括号必不可少，这是函数的必备部分。需要注意的是，该大括号后没有“;”。

(2) 关于各种变量

变量类型的定义格式为：

数据类型 变量名；

- 就基本数据类型而言，数据类型主要包括 3 类，即整型、实型和字符型。其中，整型和实型又可以再分。
- 变量名有其命名规范，可以由字母、数字等组成。
- 对变量进行定义时一定要加“;”。
- 相同类型的变量可以在一个语句中定义，此时各变量名用“,”相隔。不同类型的变量不能在一个语句中定义，否则会出现语法错误。
- 书写时，“数据类型”与“变量名”之间要留一定的空隙。

(3) 关于对变量的赋值

- 对变量所赋值必须与变量的类型相同或者相兼容，否则会出现语法错误或造成数据的丢失，所以不提倡进行不同类型数据之间的赋值。
- 虽然不同数据类型之间的赋值有可能造成数据的丢失，但是有时可利用这一特点来完成一些功能，例如实现实数的四舍五入。但是这也不是完全可靠的，最可靠的是进行强制转换。
- 赋值运算符为“=”，要与逻辑运算符“==”相区别。
- 赋值运算符的左边必须是变量，一定不能是常量。

(4) 关于运算符和表达式

- 注意自加（减）运算的计算方法， $a++$ （ $a--$ ）与 $++a$ （ $--a$ ）是不一样的。实现自加（减）的变量只能是整型、实型和字符型。常量是不能进行自加（减）运算的。
- 注意条件运算符的正确使用格式。
- 逻辑运算的结果只有两种情况，要么为“真”要么为“假”，并注意其书写格式。
- 逗号表达式的结合方式是从左到右，其值为最后一个分式的值。
- 注意几种复合运算符的书写格式。

(5) 关于程序的调试

- 静态检查：上机前首先在纸面上写好 1 个程序并且对该程序进行反复审核，确认该程序无误后再上机录入，对程序的掌握不是很熟练时最好不要凭着腹稿上机，这样不仅会浪费时间而且容易出错。
- 上机录入要认真细致，每一个符号都不要丢掉（特别注意语句后的分号）。编辑时，如果程序较长，注意录入一部分后就进行存盘操作，以免掉电丢失数据。
- 如果编译出错，系统会给出错误信息。在有误的程序行附近将有光标闪烁，有时上行的错误可能会影响到下行，从而出现错误信息，这时要注意观察分析，确实找到错误的地方再进行修改。在实际的程序调试过程中，常常出现某一处错误



改正后，再编译程序时，则另外几处的错误信息都消失了。

- 当程序运行并有了结果后，不要盲目乐观。在程序没有语法错误的情况下，还要考虑编程时算法是否也正确。检查程序的算法是否正确的有效办法是多输入几组数据，看运行结果是否正确。
- 程序调试是一项很细致的工作，多练习、多总结、多思考才能提高调试程序的技能。值得强调的是，调试程序一定要有充分的耐心，也许编写 1 个程序只需 20 分钟，但调试 1 个程序可能需要几个小时，即使熟练的编程人员也会如此。

四、改正下列错误

(1) 对于 C 程序的结构：

```
include<stdio>;  
{  
    函数体  
};
```

主要错误：头文件的使用格式，“;”的正确使用。

正确的结构为：

```
#include<stdio.h>  
{  
    函数体  
}
```

由此可见，具体的出错地方为：

- 在头文件前少了符号“#”。
- 头文件书写错误，应是“stdio.h”。
- 包含头文件是命令而不是语句，所以不应有“;”。
- 主函数结束处不应有“;”。

(2) 变量的定义：

```
int a,  
Char ch;  
floatt;  
int a;b;  
char ch,float f;
```

以上各语句的错误和正确格式如下。

- 语句应以“;”作为结束符。正确的形式为：
int a;
- 关键字书写错误。正确的形式为：
char ch;
- 数据类型与变量名之间要留空格，正确的形式为：
float t;
- 相同类型的变量一起定义时，各变量之间要用“,”相隔，而不是“;”。正确的形式为：
int a, b;