



21 世纪高职高专规划教材
高等职业教育规划教材编委会专家审定

C YUYAN CHENGXU SHEJI
SHIXUN

C语言程序设计实训

主 审	陈高锋	杨 辉
主 编	段智毅	张 淼
副主编	刘云峰	梁春华
参 编	王 超	



北京邮电大学出版社
www.buptpress.com



21 世纪高职高专规划教材

高等职业教育规划教材编委会专家审定

C 语言程序设计实训

主 审	陈高锋	
主 编	段智毅	杨 辉
副主编	刘云峰	张 森
参 编	王 超	梁春华



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书依照高等职业技术教育计算机类、电子类专业人才培养方案对 C 语言程序设计实训能力的培养要求,从职业技术教育教学改革的角度出发,以能力培养为本位,指导学生具有一定的程序设计思想,具有较强的程序设计能力。

全书共分为 11 章,以 Turbo C 为平台,内容包括:C 语言概述,C 程序设计基础,顺序结构,选择结构,循环结构,字符型数据,数组,函数,指针,结构体、共用体和用户自定义类型,文件。

本书可作为高职高专院校计算机类、电子类专业 C 语言程序设计实训的教材,也可作为各类计算机爱好者及程序设计人员的参考书。

图书在版编目(CIP)数据

C 语言程序设计实训/段智毅,杨辉主编. --北京:北京邮电大学出版社,2012.4

ISBN 978-7-5635-2923-0

I. ①C… II. ①段…②杨… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 028743 号

书 名: C 语言程序设计实训

主 编: 段智毅 杨 辉

责任编辑: 彭 楠

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京联兴华印刷厂

开 本: 787 mm×1 092 mm 1/16

印 张: 7

字 数: 166 千字

印 数: 1—3 000 册

版 次: 2012 年 4 月第 1 版 2012 年 4 月第 1 次印刷

ISBN 978-7-5635-2923-0

定 价: 16.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

高等职业院校的教育教学目标是培养技能型、应用型人才,注重学生动手能力的培养及综合实践能力的提高。围绕高职高专职业教育目标,本书旨在提高 C 语言程序设计课程的学习效果,培养学生分析问题和解决问题的能力,培养学生具有一定的程序设计思想,具有较强的程序编写能力。

本书配合已经出版的由段智毅主编的 C 语言程序设计教材使用。全书共分 11 章,每章从结构上讲又分为知识重点和编程要点以及具体的程序应用两大部分。知识重点和编程要点主要是对本章要掌握的重点知识进行归纳整理,并对编程中要注意的地方及常出问题的地方进行讲解说明;程序应用主要是围绕本章要掌握的知识点进行具体的程序编写练习。

本书由杨凌职业技术学院段智毅担任第一主编,由山东水利职业学院杨辉担任第二主编,全书由段智毅统稿。其中,第 1 章由山东水利职业学院杨辉编写,第 2、5、8 章由辽阳职业技术学院张森编写,第 3、4 章由辽宁信息职业技术学院王超编写,第 6、7 章由山西工程职业技术学院刘云峰编写,第 9、10、11 章由山西省财政税务专科学校梁春华编写。全书由杨凌职业技术学院陈高锋主审。

本书在编写过程中得到了北京邮电大学出版社有关领导和编辑的大力支持和帮助,在此表示感谢。由于编者水平有限,书中难免出现疏漏和不足之处,敬请广大读者和同仁批评指正。

编 者

目 录

第 1 章 C 语言概述	1
1.1 知识重点与编程要点	1
1.1.1 知识重点	1
1.1.2 编程重点	1
1.2 C 调试环境初识实训	1
1.2.1 实训目的和要求	1
1.2.2 实训内容和具体步骤	2
第 2 章 C 程序设计基础	9
2.1 知识重点和编程要点	9
2.1.1 知识重点	9
2.1.2 编程要点	10
2.2 C 程序基础知识实训	10
2.2.1 实训目的和要求	10
2.2.2 实训内容和具体步骤	10
第 3 章 顺序结构	15
3.1 知识重点和编程要点	15
3.1.1 知识重点	15
3.1.2 编程要点	15
3.2 顺序结构的程序设计	16
3.2.1 实训目的和要求	16
3.2.2 实训内容和具体步骤	16
第 4 章 选择结构	25
4.1 知识重点和编程要点	25
4.1.1 知识重点	25
4.1.2 编程要点	27
4.2 选择结构的程序设计	27
4.2.1 实训目的和要求	27
4.2.2 实训内容和具体步骤	28
第 5 章 循环结构	39
5.1 知识重点和编程要点	39
5.1.1 知识重点	39
5.1.2 编程要点	40
5.2 循环结构的程序设计	40
5.2.1 实训目的和要求	40
5.2.2 实训内容和具体步骤	40

第 6 章 字符型数据	47
6.1 知识重点和编程要点	47
6.1.1 知识重点	47
6.1.2 编程要点	48
6.2 字符型数据输入和输出	48
6.2.1 实训目的要求	48
6.2.2 实训内容和具体步骤	49
第 7 章 数组	50
7.1 知识重点和编程要点	50
7.1.1 知识重点	50
7.1.2 编程要点	51
7.2 数组的应用	51
7.2.1 实训目的要求	51
7.2.2 实训内容和具体步骤	51
第 8 章 函数	64
8.1 知识重点和编程要点	64
8.1.1 知识重点	64
8.1.2 编程要点	65
8.2 函数的应用	65
8.2.1 实训目的和要求	65
8.2.2 实训内容和具体步骤	66
第 9 章 指针	76
9.1 知识重点和编程要点	76
9.1.1 知识重点	76
9.1.2 编程要点	81
9.2 指针的应用	82
9.2.1 实训目的和要求	82
9.2.2 实训的内容和具体步骤	82
第 10 章 结构体、共用体和用户自定义类型	87
10.1 知识重点和编程要点	87
10.1.1 知识重点	87
10.1.2 编程要点	92
10.2 结构体和共用体的应用	93
10.2.1 实训目的和要求	93
10.2.2 实训的内容和具体步骤	93
第 11 章 文件	97
11.1 知识重点和编程要点	97
11.1.1 知识重点	97
11.1.2 编程要点	100
11.2 文件的应用	101
11.2.1 实训的目的和要求	101
11.2.2 实训的内容和具体步骤	101
参考文献	105

第 1 章 C 语言概述

1.1 知识重点与编程要点

1.1.1 知识重点

从编程的角度看,第 1 章需要掌握的知识重点包括以下内容。

(1) C 程序的构成如下。

① 程序的顶部一般要使用 `#include "stdio.h"` 语句。

② C 程序由函数构成,至少包含一个 `main` 函数,函数用一对花括号 `{ }` 括起来。

③ 定义变量。变量在使用之前一定要先定义,定义变量的语句要尽量放在程序开始的位置。

④ 通过调用 `printf` 函数显示信息在屏幕上。

(2) 要完成一个 C 程序的调试,必须经过编辑源程序、编译源程序、连接目标程序和运行可执行程序 4 个步骤。

1.1.2 编程重点

(1) C 语言严格区分大小写,因此关键字和系统函数名一定要用小写字母。

(2) 编译程序出现多个错误提示时,优先处理前面的错误,因为后面的错误很可能是由前面的错误引起的,修改前面的一个错误经常可以消除后面的大量错误。

(3) 有时候编译程序检查下一行才能发现前面的程序再现的错误,因此,错误信息给出的提示信息中的行号不一定正确,需要程序员检查前面一行甚至几行程序。

1.2 C 调试环境初识实训

1.2.1 实训目的和要求

(1) 了解 Turbo C 的调试环境,掌握进入 Turbo C 调试环境的方法。

(2) 掌握调试 C 程序的 4 个步骤:编辑、编译、链接和运行。

- (3) 熟悉 Turbo C 菜单中与调试 C 程序相关的几个命令。
- (4) 掌握 Edit 下拉菜单中的全部命令。
- (5) 掌握 C 程序中定义主函数的方法。
- (6) 掌握在 C 程序中书写注释的方法。
- (7) 了解函数 printf 的基本功能。

1.2.2 实训内容和具体步骤

1. 进入 Turbo C 调试环境

如果用户使用的是 Windows 操作系统,若 tc.exe 文件存放在 D:\tc20 目录下,可以有两种方式进入 Turbo C 调试环境。

(1) 在 Windows 环境下直接进入 Turboc C 环境。

在 Windows 环境下,双击“我的电脑”,在目录表中单击 D:\tc20,最后双击 Turbo C 系统应用程序“TC.EXE”。如图 1-1 所示。

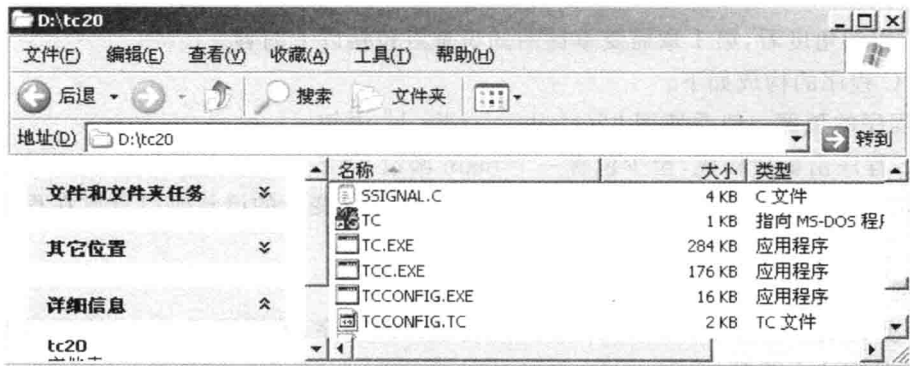


图 1-1 Windows 环境下直接进入 Turbo C 环境

(2) 在命令行方式下进入 Turbo C 环境。

在 Windows 环境下,执行“开始”→“程序”→“附件”→“命令提示符”,进入命令行方式,输入 D:,系统将当前盘符切换到 D 盘。

在当前盘符下输入 cd tc20,切换到 D 盘 tc20 目录下,最后执行“tc ↵”,进入 Turbo C 系统。如图 1-2 所示。

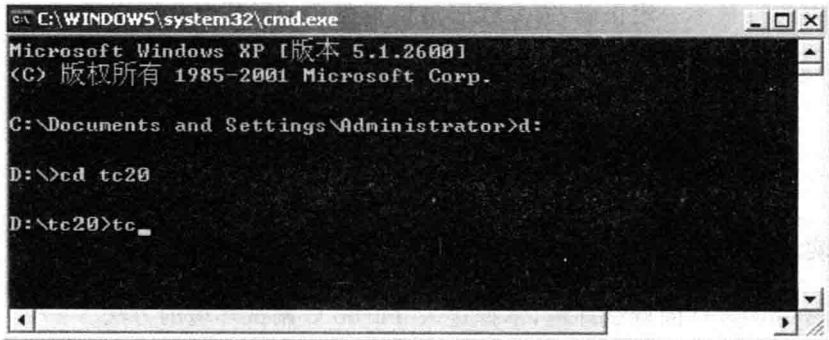


图 1-2 在命令行方式下进入 Turbo C 环境

2. 编辑程序 first.c

编辑程序是调试 C 程序的 4 个步骤中的第一步。编辑程序时,不但可以使用 Turbo C 中提供的全屏幕编辑环境,也可以使用 Windows 中的记事本或其他编辑程序。

(1) 使用 Turbo C 的全屏幕编辑环境。

在“File”菜单下选择“New”命令,在编辑区输入源程序 first.c 的全部内容,然后选择“File”菜单下的“Save”命令,并在随后的“保存文件”对话框中输入源程序文件名“first.c”,最后单击“OK”按钮。

(2) 使用 Windows 中的记事本。

在 Windows 环境下,执行“开始”→“程序”→“附件”→“记事本”,在记事本的编辑区中输入程序,如图 1-3 所示;然后,执行记事本主菜单的“文件”→“保存”命令,弹出“另存为”对话框,如图 1-4 所示,指定存储程序的目录和文件名 first.c,并将保存类型设置为“所有文件”,最后单击“保存”按钮。在 Windows 环境下生成的源程序文件,也可以使用 Turbo C 的“File”主菜单下的“Open”命令打开。

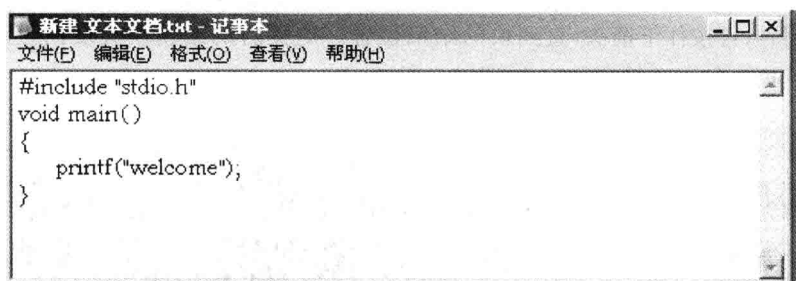


图 1-3 使用 Windows 中的记事本编辑 C 程序



图 1-4 “另存为”对话框

3. 练习使用 Turbo C 的全屏幕编辑环境中 Edit 下拉菜单中的全部命令

Edit 主菜单下的命令包括 Undo、Redo、Cut、Copy、Paste、Clear、Copy example 和 Show Clipboard, 它们都是编辑程序时需要使用的命令。

(1) 使用 Undo、Redo 命令。

Undo 命令的功能是撤销刚刚做过的编辑工作; Redo 则是取消刚刚撤销的工作, 再恢复到以前的状态。

做如下练习。

在编辑区键入一个字母“A”, 执行 Undo 命令, 那么, 刚才键入的“A”就在屏幕上消失了, 再执行 Redo 命令, “A”又重新出现在屏幕中。

(2) 使用 Cut、Copy、Paste 和 Clear 4 条编辑命令对一个编辑块进行剪切、复制、粘贴和删除。

做如下练习。

① 定义一个编辑块。定义编辑块的方法是: 用上、下箭头走到块的开始行, 然后, 在按住 Shift 键的同时, 再按住下箭头, 直到块的结束位置。定义后的编辑块会反白显示。将 first.c 的两行注释行定义为一个编辑块。如图 1-5 所示。

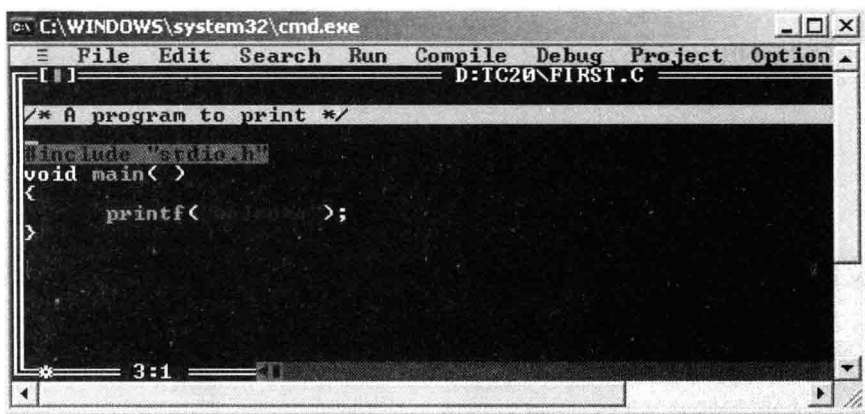


图 1-5 定义的编辑块

② 使用 Edit 菜单下的 Cut 命令将块中的文字从原来的地方删除, 这部分内容会存储在系统的剪贴板中。

③ 将光标移动到程序的最后, 使用 Paste 命令将系统的剪贴板中的文字粘贴到编辑窗口光标所在处, 则两条注释语句从程序的首部移到程序的最后。如图 1-6 所示。

④ 将 printf("welcome"); 定义为一个编辑块, 在“Edit”菜单下使用“Copy”命令, 将块中的文字复制到系统的剪贴板中。

⑤ 使用“Clear”命令将块中的文字从编辑窗口中删除。

⑥ 使用“Paste”命令将系统剪贴板中的 printf("welcome"); 语句重新粘贴到源程序中。

(3) 使用“Copy example”和“Show Clipboard”命令。

“Copy example”命令的功能是将系统 Help 中的实例复制到系统的剪贴板中。

“Show Clipboard”命令是显示系统剪贴板中的当前内容。

做如下练习。

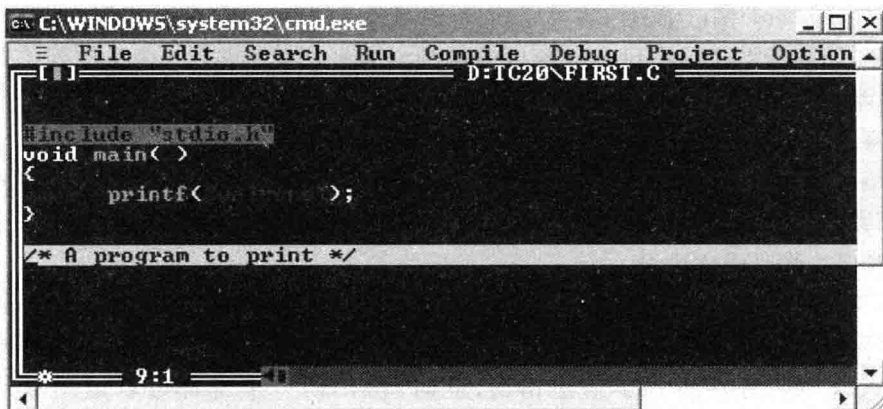


图 1-6 注释语句从程序的首部移到程序的最后

① 选择“Help”主菜单下的“Index”命令后,在“Help”窗口中输入 printf 找到该函数的使用方法,并将使用 printf 的例子显示在“Help”窗口中,然后使用“Copy example”将 printf 函数的例子复制到系统的剪贴板中。如图 1-7 所示。

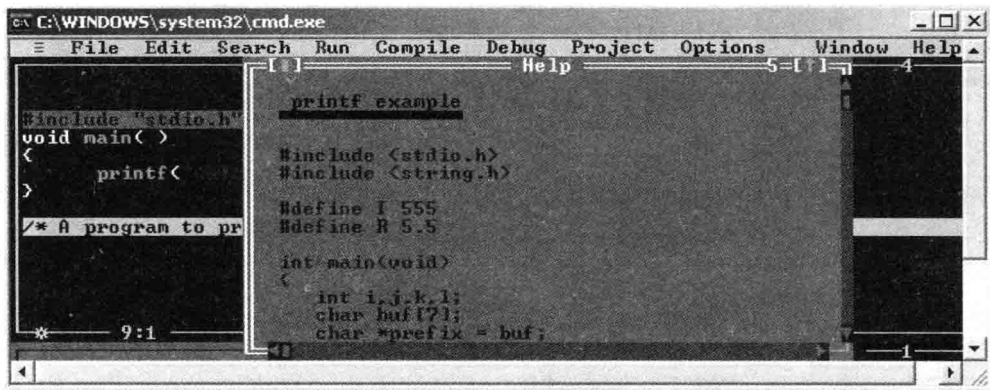


图 1-7 在系统的 Help 中调用 printf 函数的例子

② 使用“Show Clipboard”命令显示系统剪贴板中的内容,就可以在 Clipboard 窗口中看到刚才剪贴的内容。如图 1-8 所示。

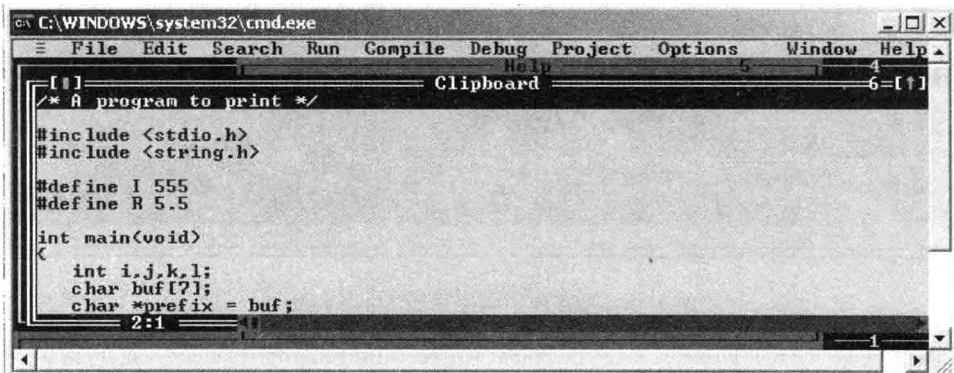


图 1-8 Clipboard 窗口中的内容

注意,“Clipboard”窗口中的内容可以编辑,但是不能直接对 Clipboard 窗口中的内容进行编译。因此,如果需要运行 Clipboard 窗口的实例,还需要将其粘贴到一个 C 源程序文件中。

4. 编译 first 程序

(1) 编译命令的使用。

如果编译源文件 first.c,首先应该保证编辑窗口中的文件是 first.c 源程序,然后在“Compile”菜单中,选择“Compile”命令,或者按快捷键 Alt+F9,程序将被编译。

(2) 发现并修改语法错误。

① 将 first.c 中的 void main()一行去掉,这样源程序就变成了一个有语法错误的程序。

② 按快捷键 Alt+F9 编译该程序。编译以后,编译窗口显示程序有一个错误(Error 后面的数字),这说明程序出现了语法错误,此时,按任何一个键都可以退出显示错误的“Message”对话框,显示出错误信息。如图 1-9 和图 1-10 所示。

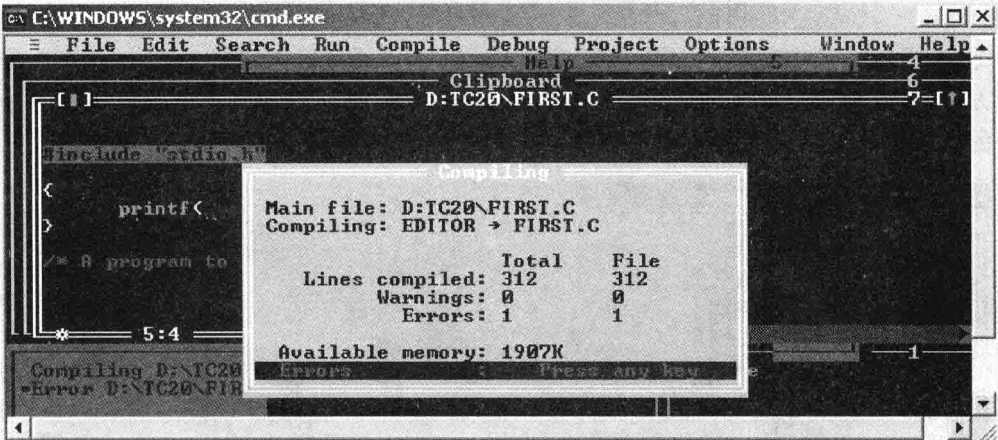


图 1-9 编译出现错误后的 Compiling 窗口

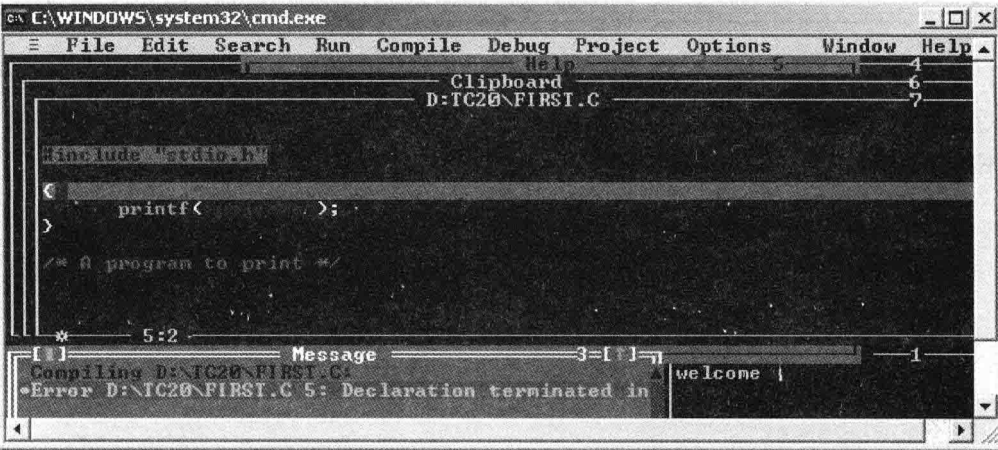


图 1-10 编译出现错误后的 Message 窗口

错误信息是“Error FIRST.C 2:Declaration terminated incorrectly”,这句话的意思是:“FIRST.C 的第 2 行有错误:声明结束错误”。“声明结束错误”是对程序语法错误的解释,

但有些时候这些解释并不能准确地告诉程序员如何修改和在哪里修改程序,需要程序员根据经验去判断错误之处,这就要求读者在调试程序过程中不断地积累经验。这里,第2行的花括号本身并没有错误,而是在它的前面缺少了一些声明函数的语句。

③ 按 Enter 键回到编辑状态,系统光标会停在程序第一个出现错误的地方,将程序修改正确,保存并再次编译。

(3) Include 目录设置。

如果在编译程序中出现了如图 1-11 所示的错误,说明 Include 目录设置不正确。Include 是 Turbo C 的一个重要目录,编译系统必须根据 Include 目录设置查找文件,如常用的 stdio.h 等。如果 Include 目录在 D 盘的 TC 目录的子目录下,而 Include 目录设置却在 C 盘的 TC 目录的子目录下,就会出现如图 1-11 所示的错误。

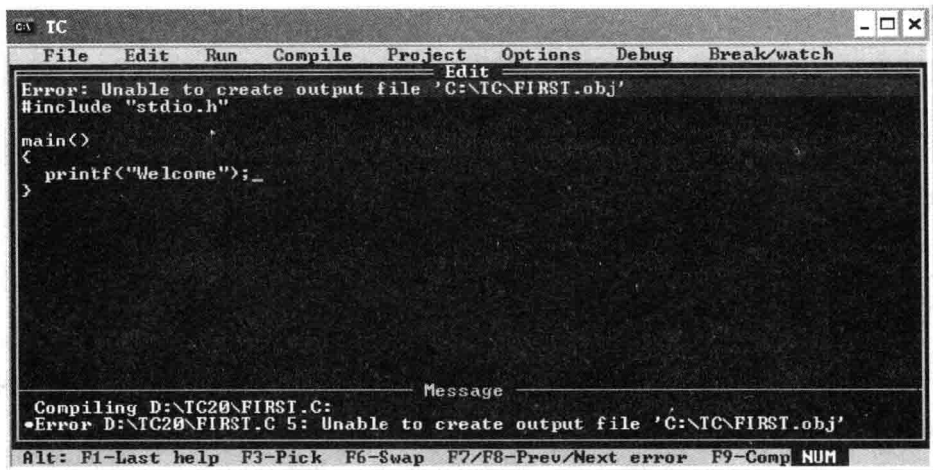


图 1-11 目录设置不正确时出现的错误

修改 Include 目录设置的方法是:使用“Options”主菜单下的“Directories”命令,调出“Directories”对话框,如图 1-12 所示。在“Include Directories”下的编辑框中设置正确的目录(同时也可以设置其他的编辑框),单击“Enter”按钮执行。

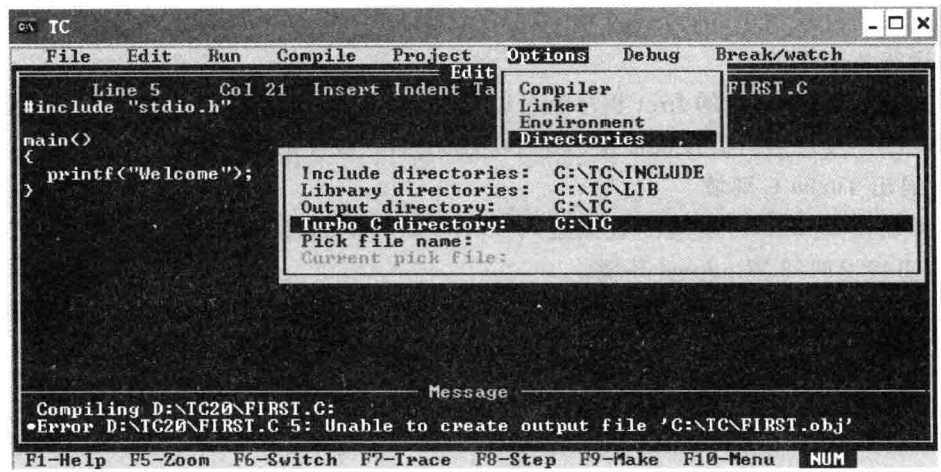


图 1-12 目录设置对话框

修改 Include 目录设置以后保存该设置,否则,每次进入 Turbo C 都要重新设置。保存 Include 目录设置的方法是:使用“Option”主菜单下的“Save”命令。

5. 链接 first 程序

(1) 链接命令使用。

如果试图链接 first 程序,首先应该保证 first.c 源程序已经成功编译。然后在“Compile”菜单中选择“Link”命令。

(2) 发现链接错误。

① 将 first.c 中的 `printf("welcome");` 改为 `print("welcome");`,使 `printf` 函数名少了一个字符 `f`。按快捷键 `Alt+F9` 编译该程序,编译成功。

② 执行“Link”命令,链接出现错误,系统使用“Message”对话框显示错误信息。如图 1-13 所示。



图 1-13 链接错误

错误信息是“Linker Error: Undefined symbol _print in module FIRST.C”,这句话的意思是:“链接错误:在 FIRST.C 模块中存在一个未定义的符号 `print`”,原因是系统中出现了一个未定义的符号。

6. 运行 first 程序

经过编译,链接成功的 first 程序就可以运行了。在“run”菜单选择“run”命令或按快捷键 `Ctrl+F9`。程序运行后按快捷键 `Alt+F5`,就会看到程序的运行情况。

7. 退出 Turbo C 环境

如果是在 Windows 环境下直接进入 Turbo C 环境的,使用“file”主菜单下的“quit”命令,可以直接退回到 Windows 环境。

如果是在命令行方式下进入 Turbo C 环境的,首先使用“file”主菜单下的“quit”命令,退到命令行方式下,然后再键入命令 `exit`,就可以退回到 Windows 环境下。

第2章 C 程序设计基础

2.1 知识重点和编程要点

2.1.1 知识重点

1. 整型常量

C 语言中的整型常量有三种形式：十进制整型常量、八进制整型常量(用数字 0 开头)和十六进制整型常量(用 0x 或 0X 开头)。

2. 整型变量

整型变量可分为基本型(int)、短整型(short int 或 short)、长整型(long int 或 long)和无符号型(unsigned)四种。

3. 实型常量

C 语言中的实型常量有两种形式：十进制数形式和指数形式。

4. 实型变量

C 语言中的实型变量分为两种：单精度类型(float)和双精度类型(double)。

5. 字符常量

C 语言的字符常量代表 ASCII 字符集里的一个字符，在程序中要用单引号括起来。

6. 字符变量

C 语言中的字符变量用关键字 char 定义，每个字符变量中只能存放一个字符。

7. 字符串常量

C 语言中的字符串常量是由一个双引号括起来的字符序列。

8. 算术运算符与算术表达式

算术运算符有加(+)、减(-)、乘(*)、除(/)、取余运算(%)。其中，取余运算的对象只能是整型，其余运算符的运算对象可以是整型，也可以是实型。

用算术运算符和一对圆括号将运算量连接起来的、符合 C 语言语法的表达式称为算术表达式。运算对象可以是常量、变量和函数等。

9. 赋值运算符与赋值表达式

在 C 语言中，“=”符号称为赋值运算符，由赋值运算符组成的表达式称为赋值表达式。

10. 自加、自减运算符和逗号运算符

自加运算符“++”和自减运算符“--”的运算结果是使运算对象的值增 1 或减 1。运

算对象可以是整型变量,也可以是实型变量,不能是常量或表达式。

逗号运算符“,”是 C 语言提供的一种特殊运算符,用逗号将表达式连接起来的式子称为逗号表达式。

2.1.2 编程要点

(1) 能够正确定义并使用常用数据类型,如整型、实型、字符型,知道不同数据类型的使用范围。

(2) 掌握五种基本算术运算符的使用: +、-、*、/、%, 知道优先级与结合性。

(3) 掌握赋值运算符和赋值表达式的正确用法。

2.2 C 程序基础知识实训

2.2.1 实训目的和要求

(1) 掌握 C 语言数据类型,熟悉如何定义一个整型、字符型和实型变量,以及对它们赋值的方法。

(2) 在混合运算中,不同的数据类型由系统自动实现转换,由少字节类型向多字节类型转换;也可由强制转换运算符完成转换。

(3) 学会使用 C 语言的有关算术运算符,以及包含这些运算符的表达式。

(4) 掌握自加“++”和自减“--”运算符的使用。

(5) 进一步熟悉 C 程序的编辑、编译、连接和运行过程。

2.2.2 实训内容和具体步骤

1. 调试程序,分析输出结果。

(1) 输入并运行以下程序。

```
main()
{
    unsigned int a,b;
    a = 32767;
    b = a + 10;
    printf("a = %u,b = %u\n",a,b);
}
```

将程序改为

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



```

a = 32767;
b = a + 10;
printf("a = %d,b = %d\n",a,b);
}

```

说明:int 型变量只能容纳 $-32\,768 \sim +32\,767$ 范围内的数,超出此范围将产生溢出,输出结果错误,但运行时并不报错。

(2) 输入并运行以下程序。

```

main()
{
    float a,b;
    a = 123456.789e5;
    b = a + 20;
    printf("a = %f,b = %f\n",a,b);
    printf("a = %e,b = %e\n",a,b);
}

```

说明:float 型变量最多只能保存 7 位有效数字,后面的数字是无意义的,不能准确表示该数。因此,应避免用一个很大的实数和一个很小的实数相加减,否则会丢失小的数。

若要得到一个正确的 b 值,应该怎样改写上面程序?

(3) 输入并运行以下程序。

```

main()
{
    char c1,c2;
    c1 = 'a';
    c2 = 'b';
    printf("%c, %c\n",c1,c2);
}

```

① 增加语句 `printf("%d, %d\n",c1,c2);`并运行之。

② 将第三行改写为 `int c1,c2;`再运行之。

③ 将第三行改写为 `c1=400; c2=500;`再运行之,分析其运行结果。

说明:字符型数据可作为整型数据处理,整型数据也可以作为字符型数据处理,但要注意,由于字符型数据只占一个字节,它只能存放 $0 \sim 255$ 范围内的整数。

(4) 输入并运行以下程序。

```

main()
{
    int a,b;
    b = (a = 3,7 * 3);
    printf("(1).b = %d, a = %d\n",b,a);
    b = a = 3,7 * a;
    printf("(2).b = %d, a = %d\n",b,a);
}

```