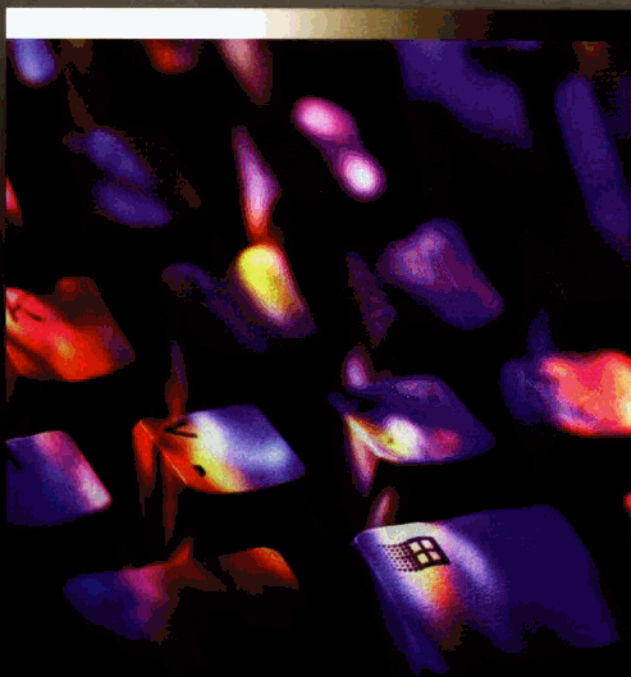




姚庭宝 封孝生 编著

C语言

程序设计习题与解析



12C
d1

国防科技大学出版社

前 言

大学本科生学习计算机第一教学语言是他们在大学期间培养训练使用计算机语言及应用能力的重要一环。《计算机程序设计语言》课程作为讲授计算机第一教学语言进行安排。该课程的教学目的是使学生较系统地学习一种优秀、实用且应用广泛的计算机程序设计语言——C语言,熟悉它的基本概念、各种类型的数据结构与语句特性,了解掌握用C语言进行程序设计的基本方法与技巧,促进养成良好的编程风格和习惯,锻炼和提高使用计算机解决应用问题的能力,为今后进一步的程序设计实践打下较扎实的基础。国防科技大学出版社2001年2月出版的《C语言及编程技巧》一书作为《计算机程序设计语言》课程的教材,已在包括国防科技大学在内的一些高等院校使用,获得广大学生的欢迎。

对于广大学生,熟悉一种计算机程序设计语言的最好方法是阅读理解各类程序示例,结合应用实际多编写源程序并经上机调试通过。编程上机综合了对计算机程序设计语言的理解、对编程工具与环境的操作以及对程序设计方法与技巧的运用。为了帮助广大学生进一步培养、检验他们阅读理解程序示例、独立编程以及上机操作的能力,提高并加强学生计算机应用的整体水平,我们精心设计并编写了本书。本书可起到配套教学参考书和习题集的双重作用。本书亦可单独使用,是一本学用C语言进行程序设计的指导读物。

全书由十个单元和两个附录组成。这十个单元是:Turbo C 2.0 集成开发环境上机指南;顺序结构程序设计;选择结构程序设计;循环结构程序设计;数组与字符串;函数;指针;结构体与线性链表;文件;共用体、枚举类型、位运算、编译预处理及通用程序设计。除第一单元外,其余九个单元与《C语言及编程技巧》的相应章相呼应,每一单元内包括知识要点、选择题、填空题、程序分析题、程序设计题、参考解答与解析六大部分。附录一是C程序常见错误示例,列出了在C语言编程中各类常见错误,配以示例程序,并说明其出错原因及改正方法。附录二是C语言程序设计试题选,这是专为那些对C语言程序设计有强烈兴趣并学有余力者设计的一份“套餐”,该附录中提供了一批经精选较为典型、具有一定技巧性和趣味性,且富于启发性的C语言程序设计试题,这些程序设计试题本身具有一定的挑战性。编者鼓励学生对其中一些试题编写C程序并上机调试通过,藉以进一步发展程序设计技巧,提高灵活运用C语言解决应用问题的本领,并深入体现C语言程序设计的潜在能力。

本书力求体现下列特色:内容翔实,题型丰富,信息量大,覆盖面广,分析透彻,解析准确,应用面广,实用性强。

衷心感谢各自的亲人杜秀荣女士、刘晓女士对我们工作的一贯支持。国防科技大学出版社何晋不仅在本书的编著过程中提出了不少中肯的建议,还帮助润色了文字。国防

科技大学出版社领导一直关心和支持本书的出版,不少工作人员付出了辛勤的劳动。在此一并致谢。

对于本书的内容选择和文字综述上的不当之处,热忱欢迎广大读者提出批评、建议。

国防科技大学人文与管理学院

姚庭宝* 封孝生

2002年6月

* 联系地址:湖南省长沙市国防科技大学人文与管理学院,邮编 410073,电话:(0731)4575557(H),4575578(O)。

目 录

第一单元 Turbo C 2.0 集成开发环境上机指南	(1)
1.1 Turbo C 2.0 的安装和启动	(1)
1.2 Turbo C 2.0 的集成开发环境	(1)
1.3 Turbo C 2.0 的配置文件	(7)
1.4 C 程序的开发和运行	(8)
1.5 C 程序的调试	(8)
第二单元 顺序结构程序设计	(10)
2.1 知识要点	(10)
2.2 选择题	(11)
2.3 填空题	(13)
2.4 程序分析题	(15)
2.5 程序设计题	(18)
2.6 参考解答与解析	(19)
第三单元 选择结构程序设计	(25)
3.1 知识要点	(25)
3.2 选择题	(25)
3.3 填空题	(32)
3.4 程序分析题	(35)
3.5 程序设计题	(38)
3.6 参考解答与解析	(40)
第四单元 循环结构程序设计	(48)
4.1 知识要点	(48)
4.2 选择题	(49)
4.3 填空题	(54)
4.4 程序分析题	(59)
4.5 程序设计题	(64)
4.6 参考解答与解析	(66)

第五单元 数组与字符串	(75)
5.1 知识要点	(75)
5.2 选择题	(76)
5.3 填空题	(79)
5.4 程序分析题	(83)
5.5 程序设计题	(86)
5.6 参考解答与解析	(89)
第六单元 函 数	(102)
6.1 知识要点	(102)
6.2 选择题	(103)
6.3 填空题	(111)
6.4 程序分析题	(117)
6.5 程序设计题	(121)
6.6 参考解答与解析	(123)
第七单元 指 针	(135)
7.1 知识要点	(135)
7.2 选择题	(136)
7.3 填空题	(144)
7.4 程序分析题	(151)
7.5 程序设计题	(155)
7.6 参考解答与解析	(157)
第八单元 结构体与线性链表	(168)
8.1 知识要点	(168)
8.2 选择题	(170)
8.3 填空题	(177)
8.4 程序分析题	(183)
8.5 程序设计题	(187)
8.6 参考解答与解析	(188)
第九单元 文 件	(198)
9.1 知识要点	(198)
9.2 选择题	(199)
9.3 填空题	(202)
9.4 程序分析题	(205)

9.5 程序设计题	(209)
9.6 参考解答与解析	(210)
第十单元 共用体、枚举类型、位运算、编译预处理及通用程序设计	(220)
10.1 知识要点	(220)
10.2 选择题	(222)
10.3 填空题	(227)
10.4 程序分析题	(230)
10.5 程序设计题	(234)
10.6 参考解答与解析	(236)
附录一 C 程序常见错误示例	(246)
附录二 C 语言程序设计试题选	(270)
主要参考书目	(276)

第一单元 Turbo C 2.0 集成开发环境上机指南

1.1 Turbo C 2.0 的安装和启动

Turbo C 2.0 的安装非常简单,只要将 1# 盘插入 A 驱动器中,在 DOS 的“A: \>”下键入:

```
A:\>install
```

即可。此时屏幕上显示三种选择:

- ① Install Turbo C on a Hard drive (在硬盘上安装 Turbo C)
- ② Update Hard Drive copy of Turbo C 1.5 to Turbo C 2.0 (升级 Turbo C 1.5 版到 2.0 版,这样的安装将保留原来对选择项、颜色和编辑功能键的设置)
- ③ Install Turbo C to a floppy Drive (为只有两个软盘而无硬盘的系统安装 Turbo C 2.0)

假定按第一种选择进行安装。按回车键确定后,依次设置好 Turbo C 及其头标文件、库文件、例子程序等的安装位置,选择“Start Installation”项开始正式安装,按照系统提示,依次插入各个盘号的软盘,就可顺利地安装完毕。

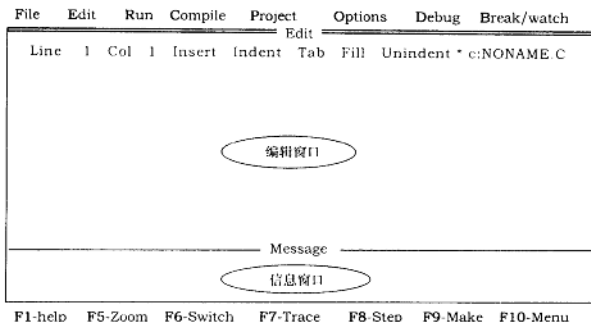
安装完毕后,即可运行 Turbo C 2.0 系统,运行时,在 Turbo C 2.0 目录下键入 TC (英文大、小写随意)即可进入 Turbo C 2.0 集成开发环境。如果在其他目录下运行 Turbo C 2.0 系统,需要写上完整的路径名或在 PATH 命令中指定 Turbo C 2.0 路径。

1.2 Turbo C 2.0 的集成开发环境

进入 Turbo C 2.0 集成开发环境后,屏幕显示如下页所示。其中最上面一行为 Turbo C 2.0 主菜单,用来进行功能项选择,中间的窗口为编辑窗口,用来显示和编辑源程序,接下来的窗口是信息窗口,用来显示系统编译和连接等提示信息,最底下一行为参考行,用来给出一些常用功能和菜单项的快捷键。这四个窗口构成了 Turbo C 2.0 的主屏幕,以后的编程、编译、调试以及运行都将在这个主屏幕中进行。

编辑窗口的顶行是编辑状态指示行,包括当前光标的行号和列号、自动缩进指示、插入/替换指示以及当前正在编辑程序的文件名等。

主菜单中除“Edit”菜单项外,其他各菜单项均有子菜单,只要用“Alt”键加上相应菜单项中第一个字母(即带黑体的大写字母),就可进入该菜单项的子菜单中。下面详细介绍主菜单的内容。



1.2.1 File(文件)菜单项

按 Alt + F 可进入 File 菜单项,该菜单项包括以下内容:

Load(加载)

装入一个存在的程序源文件到编辑窗口进行编辑,可用通配符(如 *.C)来进行列表选择。也可装入其他扩展名的文件,只要给出文件名(或只给路径)即可。该菜单项的热键为 F3 键,即只要按 F3 键即可直接进入该菜单项,而不需要先进入 File 菜单项再选此菜单项。

Pick(选择)

将最近装入编辑窗口的 8 个文件列成一个表让用户选择,选择后将该程序装入编辑区,并将光标置在上次修改过的地方。热键为 Alt + F3。

New(新文件)

关闭编辑窗口中的文件,新建一个空文件,缺省文件名为 NONAME.C。若当前编辑窗口中的文件有修改,系统会提示是否保存。

Save(存盘)

将编辑窗口中的文件存盘,若文件名是 NONAME.C 时,将询问是否更改文件名,其热键为 F2。

Write to(写盘)

将当前编辑窗口中的文件以新的文件名进行存盘,相当于“另存为”。

Directory(目录)

显示当前工作目录及目录中的文件,并可由用户选择。

Change dir(改变目录)

显示当前工作目录,用户可以改变用户的当前工作目录。

Os shell(暂时退出)

暂时退出 Turbo C 2.0 到 DOS 提示符下,此时可以运行 DOS 命令。若想返回 Turbo C 2.0 中,只要在 DOS 提示符下键入 EXIT 或 _ (英文大、小写随意)即可。

Quit(退出)

退出 Turbo C 2.0, 返回到 DOS 操作系统中, 其热键为 Alt + X。

说明: 以上各项可用光标键移动光棒进行选择, 按回车键则执行。也可用每一项带黑体的大写字母直接选择。若要退出主菜单或从它的下一级菜单项返回到上一级菜单项, 按 Esc 即可。Turbo C 2.0 所有菜单均采用这种方法进行操作, 以下不再说明。

1.2.2 Edit(编辑)菜单项

按 Alt + E 可进入编辑菜单项, 若再按回车键, 则光标出现在编辑窗口, 此时用户可以进行文本编辑。可按 F1 键获得有关编辑方法的帮助信息。Turbo C 2.0 在编辑文件时还有一种自动缩进功能, 即回车换行时光棒定位和上一行第一个非空字符对齐。与编辑有关的常用功能键如下:

F1: 获得 Turbo C 2.0 编辑命令的帮助信息

F5: 扩大编辑窗口到整个屏幕

F6: 在各窗口之间进行切换

F10: 从编辑窗口进入主菜单

PageUp / PageDown: 向前 / 向后翻页

Home / End: 将光标移到所在行的开始 / 结尾

Ctrl + Y: 删除光标所在的一行

Ctrl + T: 删除光标所在处的一个单词

Ctrl + N: 在光标所在位置插入一个回车换行符

Ctrl + KB: 标记块首

Ctrl + KK: 标记块尾

Ctrl + KV: 移动块

Ctrl + KC: 拷贝块

Ctrl + KY: 删除块

Ctrl + KH: 隐藏 / 显示块阴影

Ctrl + F1: 如果光标所在处为 Turbo C 2.0 标准库函数, 则显示该函数的帮助信息

Ctrl + OL: 自动缩进开关键

1.2.3 Run(运行)菜单项

按 Alt + R 可进入 Run 菜单项, 该菜单项包括以下内容:

Run(运行程序)

运行经源程序编译连接后生成的可执行文件, 有关生成的可执行文件名的描述见 1.2.4 中 Make EXE file 项。若其中的源代码自上次编译后未做过修改则直接运行, 否则先进行编译、连接后再运行。运行时遇到设置的断点自动暂停。热键为 Ctrl + F9。

Program reset(程序重启)

强行终止当前调试的程序, 释放分给程序的空间。热键为 Ctrl + F2。

Goto cursor(运行到光标处)

选择该项可使程序运行到光标所在行后自动暂停下来。光标所在行必须为一条可执行语句,否则提示错误。该菜单项主要在调试程序时使用。热键为 F4。

Trace into(跟踪进入)

用于程序的单步执行,在执行一条调用用户自定义的函数时,跟踪到该函数内部去执行。热键为 F7。

Step over(单步执行)

用于程序的单步执行,但不跟踪函数内部调用。热键为 F8。

User screen(用户屏幕)

显示程序运行时在屏幕上显示的结果。热键为 Alt + F5。

1.2.4 Compile(编译)菜单项

按 Alt + C 可进入 Compile 菜单项,该菜单项包括以下内容:

Compile to OBJ(编译生成目标码)

将一个 C 源文件编译生成 OBJ 目标文件。热键为 Alt + F9。

Make EXE file(生成可执行文件)

生成可执行文件,并在信息窗口显示生成的可执行文件名。其中可执行文件名是下面几项之一:

- ①由 Project/Project name 指定的项目名。
- ②若没有项目名,则由 Primary C file 指定的源文件。
- ③若以上两项都没有指定,则为当前窗口的源文件名。

Link EXE file(连接生成可执行文件)

把当前 OBJ 文件及库文件连接在一起生成可执行文件。可执行文件名见上面 Make EXE file 项说明。

Build all(建立所有文件)

重新编译项目里的所有文件,并生成可执行文件。该命令不作时间过时检查(上面的几条命令要作时间过时检查,即如果目前项目里源文件的日期和时间与目标文件相同或更早,则不对源文件进行编译)。

Primary C file(主 C 文件)

当在该项中指定了主文件后,在以后的编译中,如没有项目文件名则编译此项中规定的主 C 文件,如果编译中有错误,则将此文件调入编辑窗口,不管目前窗口中是不是主 C 文件。

Get info(获取信息)

获得有关当前路径、源文件名、源文件大小、编译中的错误数目、可用空间等信息。

1.2.5 Project(项目)菜单项

按 Alt + P 可进入 Project 菜单项,该菜单项包括以下内容:

Project name(项目名)

项目名是以 .PRJ 作为扩展名的文件名,其中包括将要编译、连接的文件名。例如有

一个程序由 file1.c、file2.c、file3.c 组成,要将这3个文件编译连接成一个 file.exe 的可执行文件,可以建立一个项目名为 file.prj 的项目文件,其内容如下:

```
file1.c  
file2.c  
file3.c
```

此时将 file.prj 放入 Project name 项中,以后进行编译时将自动对项目文件中规定的三个源文件分别进行编译,然后连接成 file.exe 文件。

当项目文件中的文件无扩展名时,均按源文件对待。另外,项目文件中的文件也可以是库文件,但必须写上扩展名.LIB。

Break make on(中止编译)

这是个多项选择开关。由用户选择是否有 Warning(警告)、Errors(错误)、Fatal errors(致命错误)时或 Link(连接)之前退出 Make 编译。

Auto dependencies(自动依赖)

这是个“跟斗式”开关。当开关置为 on,编译时将检查源文件与对应目标文件的日期和时间是否一致,而当开关置为 off 时,则不进行检查。

Clear project(清除项目名)

清除 Project/Project name 中的项目名。

Remove messages(删除信息)

把错误信息从信息窗口中清除掉。

1.2.6 Options(选项)菜单项

按 Alt + O 可进入 Options 菜单项,初学者要谨慎使用该菜单项,它包括以下内容:

Compiler(编译器)

本项选择又有许多子菜单项,可以让用户选择编译模式、硬件配置、存储模型、调试技术、代码优化、对话信息控制和宏定义等。这些子菜单如下:

Model:编译模式。共有 tiny、small、medium、compact、large、huge 六种不同模式可由用户选择。

Define:全局宏定义。打开一个宏定义框,供用户输入宏定义,多个宏定义之间用分号分隔。

Code generation:与代码生成有关的设置。

Optimization:程序优化设置选项,设定是优化程序的大小还是优化程序的速度、是否允许使用寄存器变量等。

Source:与源程序有关选项,设定标识符的有效字符的个数、源程序中是否允许嵌套注释以及是否只使用 ANSI C 关键字等。

Error:编译过程中对编译错误和编译警告的处理设置。

Names:用于改变段(segment)、组(group)和类(class)的名字,一般不作修改。

Linker(连接器)

本菜单设置有关连接的选择项,它包括以下内容:

Map file:选择是否产生 MAP 文件及 MAP 文件的类型。

Initialize segments:是否在连接时初始化没有初始化的段。

Default libraries:是否在连接其他编译程序产生的目标文件时去寻找其缺省库。

Graphics library:是否连接图形库中的函数。

Warn duplicate symbols:当有重复符号时产生警告信息。

Stack warning:是否让连接程序产生 Nostack 的警告信息。

Case-sensitive link:连接时标识符是否区分大、小写。

Environment (环境)

本菜单项对编辑环境进行设置,如是否文件自动存盘、制表键和屏幕大小等。

Message tracking:对消息窗口中的错误和警告信息的跟踪方式。

Keep message:编译前是否清除消息窗口中的信息。

Config auto save:指定是否自动保存有关配置信息。选 on 时,在 Run、Os shell 或退出集成开发环境之前,如果 Turbo C 2.0 的配置被改过,则所做的改动将存入配置文件中。选 off 时不存。

Edit auto save:是否在 Run 或 Os shell 之前,自动存储编辑的源文件。

Backup file:是否在源文件存盘时产生后备文件(.BAK 文件)。

Tab size:设置制表键大小,默认为 8。

Zoomed windows:将现行活动窗口放大到整个屏幕,热键为 F5。

Screen size:设置屏幕文本大小。

Directories (路径)

设置头标文件、库文件等的路径,包括下列各项:

Include directories:头标文件的路径,多个路径用“;”分开。

Library directories:库文件路径,多个路径用“;”分开。

Output directory:输出文件(OBJ、EXE、MAP 文件)的目录。

Turbo C directories:Turbo C 所在的路径。

Pick filename:定义 pick 文件名。

Arguments (命令行参数)

允许用户设置命令行参数。

Save options (存储配置)

保存所设置的编译、连接、路径等信息到配置文件中,缺省的配置文件为 TCCONFIG.TC。

Retrive options (恢复选项)

装入一个配置文件到 TC 中,TC 将使用该文件的设置项。

1.2.7 Debug (调试) 菜单项

按 Alt + D 可选择 Debug 菜单项,该菜单主要用于调试程序,它包括以下内容:

Evaluate (求值)

在单步调试或断点调试中使用,用来查看变量或表达式的值,并可用来修改变量的值。

Call stack(调用栈)

显示当前已调用而尚未返回的函数调用序列,主函数在栈底,正在运行的函数在栈顶。

Find function(定位函数)

在运行 Turbo C Debugger 时用于显示规定的函数。

Refresh display(恢复显示)

如果编辑窗口偶然被用户窗口重写了,可用此恢复编辑窗口的内容。

1.2.8 Break/watch(断点及监视)菜单项

按 Alt + B 可进入 Break/watch 菜单项,该菜单项包括以下内容:

Add watch(增加监视项)

增加一个需要监视的变量或表达式,设定后在屏幕底部出现一个监视窗口。单步或断点调试时,将动态显示监视的变量或表达式的值。

Delete watch(删除监视项)

从监视窗口中删除一个监视变量或表达式。

Edit watch(编辑监视项)

编辑一个监视窗口中的监视变量或表达式。

Remove all watches(删除所有监视项)

从监视窗口中删除所有的监视变量或表达式。

Toggle breakpoint(切换断点)

对光标所在的行设置或清除断点。

Clear all breakpoints(清除所有断点)

清除所有断点。

View next breakpoint(查看下一断点)

将光标移动到下一个断点处。

1.3 Turbo C 2.0 的配置文件

所谓配置文件是包含 Turbo C 2.0 有关信息的文件,其中包括编译、连接的设置和路径等信息。

可以用下述方法建立 Turbo C 2.0 的配置:

(1) 建立用户自命名的配置文件

可以从 Options 菜单中选择 Options/Save options 命令,将当前集成开发环境的所有配置存入一个由用户命名的配置文件中。下次启动 TC 时只要在 DOS 下键入:

tc/c <用户命名的配置文件名> -j

就会按这个配置文件中的内容作为 Turbo C 2.0 的选择。

(2) 若设置 Options/Environment/Config auto save 为 on,则退出集成开发环境时,当前的设置会自动存放到 Turbo C 2.0 配置文件 TC\CONFIG.TC 中。Turbo C 在启动时会自动寻找这个配置文件。

(3) 用 TC\INST 设置 Turbo C 的有关配置,并将结果存入 TC.EXE 中。Turbo C 在启动时,若没有找到配置文件,则取 TC.EXE 中的缺省值。

1.4 C 程序的开发和运行

Turbo C 2.0 提供一个集成开发环境(Integrated Development Environment, IDE),在该系统中可以完成一个 C 程序开发和运行的所有步骤。用 Turbo C 2.0 来开发和运行一个 C 程序步骤如下:

编辑 在 Turbo C 2.0 的编辑窗口中编辑 C 源文件,或可以打开一个已经存在的 C 源文件进行编辑,编辑完成后将文件存盘,存盘时缺省扩展名为“.c”。

编译 源文件编辑完毕后使用编译器对 C 语言源程序进行语法检查和翻译,生成扩展名为“.obj”的目标文件。在正式编译前需经过编译预处理。若程序有语法错误,则在屏幕底部的信息窗口中显示错误和警告信息。此时可根据提示的错误和警告信息进行修改,修改完后再,再重复进行编译。热键为 Alt + F9。

连接 程序经编译程序检查无误后,连接程序将目标文件和相关标准库文件连接在一起,形成一个扩展名为“.exe”的可执行程序。热键为 F9。

运行 生成可执行程序后,便可在 Turbo C 的集成开发环境中通过 Run 命令来运行,也可脱离 Turbo C 的集成开发环境在操作系统提示符下键入该可执行程序的文件名来直接运行。可通过运行程序来检查程序中是否存在错误,尤其是逻辑错误。若有错误,便需重新对源文件进行编辑、编译、连接、运行,直到改正程序中的错误,获得预期的正确结果时为止。

1.5 C 程序的调试

编写完一个程序只能说是完成了任务的一半甚至更少,程序中不可避免地会存在一些错误,需要在编译阶段甚至在运行阶段来查找并改正错误,从而最终完成任务。程序中的错误有两类:一类是语法错误,这是由于程序设计者对 C 语法规则的各个细节和限制不完全了解或编程时粗心大意而导致;另一类是逻辑错误,系指程序编译、连接均无错误,但程序的结果与预期不符,这主要是由于对问题的理解和把握不充分或者在算法设计中存在某些缺陷和漏洞而导致。

所谓程序调试是指对程序的查错和排错。程序调试的第一步是调试一般性语法错误,使程序能够编译通过;第二步是排除编译系统不能检查出的逻辑性语法错误,即那些由于对 C 语言的规则没有完全理解或一时疏忽而没有能将算法完全正确地转换成 C 语句的错误;第三步是排除纯逻辑错误,即由于算法或设计本身的错误或缺陷而造成的错误。第二步和第三步在具体的程序调试过程中没有明显的界限,往往是结合起来进行。

排除一般性语法错误比较简单,只需要根据编译系统提示的错误信息具体找出程序中的错误并改正之即可。但要注意以下几点:

第一,编译系统提示的出错行有时并不是真正出错的行,需要根据前后语句的关系稍作一些分析。如果在提示的出错行上找不到错误的话,就应当到上一行甚至再往上去寻找。

第二,编译系统提示的错误信息有时非常多,应当从上到下逐一改正。一大片错误信息往往使人感到问题严重,不知所措,无从下手,但这一般并不意味着程序中就有大量的错误,其实很可能仅有少许错误甚至就一两个错误。比如,对所使用的变量未加说明或说明不合法,编译系统就会对所有含该变量的程序行给出出错信息,只要加上一个语法正确的变量说明,则当重新编译时与该变量有关的所有错误行就都会消除。

第三,系统有时提示的出错类型并非绝对准确,不要死抱住提示的出错信息不放,这样容易钻牛角尖,从而不能及时解决问题。

由于出错的情况繁多且各种错误互相关联,因此要善于分析,努力找出真正的错误所在。

排除逻辑性语法错误和纯逻辑性错误有时是既艰巨又复杂,需要仔细检查和分析才能发现。要完全排除逻辑上的错误则更为困难,尤其是当程序本身较复杂时。坚持良好的程序设计风格有助于程序调试。比如,选用结构化程序设计方法,不仅可以增加程序的可读性,而且对程序的调试也非常有利。又如,在程序中加入适当的注释,可以有助于帮助理解每个程序段的作用。对于规模较大的程序,可以先将每一个模块抽取出来单独调试,确认它们都能正确完成其特定功能后,再组合起来进行系统调试。

具体调试过程中可以采用以下办法:

(1) 将程序与流程图(或伪代码)仔细对照,如果流程图是正确的而程序写错了,这类逻辑型语法错误是很容易发现的。如复合语句忘记写花括号,只要一对照流程图就能很快发现。

(2) 输入特定数据,采取“分段检查”的方法。在程序不同位置设几个 `printf` 语句,输出有关变量的值,逐段往下检查,直至找到某一段中数据不正确为止,以便将错误定位在某一个局部的程序段中。不断缩小“查错区”,就易于发现错误所在。

(3) 如果在程序中没有发现问题,而运行结果却与预期不符,那就要检查流程图有无错误,即算法有无问题,如有则改正之,并修改相应的源程序。

(4) 借助于系统提供的 `debug`(调试)工具,如单步执行、设置断点、变量监视等,跟踪流程并给出相应信息从而找出错误所在。

总之,程序调试是一项细致而深入的工作,需要下功夫、动脑子,善于积累经验,了解一些 C 程序常见的错误信息。

第二单元 顺序结构程序设计

2.1 知识要点

2.1.1 基本内容

- C 程序的基本结构
- 基本字符集与标识符
 - 基本字符集
 - 标识符
 - 关键字与标准标识符
 - 常量与变量
 - 常量标识符
- 简单数据类型
 - 整数类型
 - 实数类型
 - 字符类型
- 运算符与表达式初探
 - 算术运算符与算术表达式
 - 关系运算符与关系表达式
 - 逻辑运算符与逻辑表达式
 - 类型转换
- 赋值语句及简单输入、输出
 - 赋值语句
 - 赋值表达式
 - 格式输出语句
 - 格式输入语句
- 程序设计风格

2.1.2 小结

本单元涉及 C 语言的一些基础知识,包括 C 程序的基本结构,基本字符集与标识符,简单数据类型,运算符与表达式初探,赋值语句及简单输入、输出等。它们在 C 语言编程中将经常用到。因为一个 C 程序大体上可分为输入数据、计算或处理数据、输出数据和文字信息三大部分,而这些都往往离不开诸如常量定义、变量说明以及使用输入、输出标

准库函数和表达式、赋值语句等。另外,本单元专门把程序设计风格列为一项基本内容,在此特指编程风格。在编程中应当重视良好编程风格和习惯的养成,若能坚持不懈,对今后的编程、调试以及确保程序的正确性、可理解性、可测试性、可维护性、可重用性等方面必将大有裨益。就本单元所涉及的一些基本内容而言,可提出如下一些关于编程风格所应遵循的准则,包括:选用合适的常量标识符;选取有实际含义的标识符作为变量名;坚持按一定的缩进规则书写和键入程序;适当使用注释;使程序具有一定的交互性;贴切地安排输出格式;适当地设计测试用例,并带有预期结果等。读者可以根据自身特点概括出若干良好的编程风格,并在程序设计中坚持使用,不断充实、完善。

2.2 选择题

(说明:本书中的选择题均是从四个选项中选一。)

【题1】 以下说法中不正确的是()。

- ①C 程序是一种函数结构,函数是 C 程序的基本构件
- ②运行 C 程序总是从第一个定义的函数开始
- ③任何 C 程序必须有一个名为 main 的主函数
- ④运行 C 程序总是从 main 函数开始

【题2】 以下不是 C 语言规定保留字的是()。

- ①float ②integer ③signed ④unsigned

【题3】 以下各选项中,均是 C 语言保留字的组是()。

- ①integer, float, double ②if, else, switch
- ③struct, union, enum ④auto, static, external

【题4】 以下符合语法规则的用户标识符是()。

- ①1st ②break ③B757 ④Two Words

【题5】 以下选项中属于 C 语言的数据类型是()。

- ①复数型 ②逻辑型 ③双精度型 ④集合型

【题6】 以下选项中属于 C 语言的简单数据类型是()。

- ①整型、实型、布尔型 ②整型、实型、字符型
- ③整型、实型、字符型、字符串型 ④整型、实型、字符型、布尔型

【题7】 在 C 语言中,整常量 0238 是()。

- ①八进制数 ②十进制数 ③十六进制数 ④非法数

【题8】 在以下字符序列中,表示合法的长整常量是()。

- ①"EIDGFE" ②EIDGFE ③0L ④3.14159265358

【题9】 以下诸运算符中优先级为最高和最低的运算符是()。

&&, ||, =, ==, !, ~,

- ①"!"和"~" ②"!"和"=" ③"&&"和"==" ④"&&"和"="

【题10】 为说明 int 型变量 i、j、k 并均赋以初始值 0,下列诸变量说明语句中正确的是()。