

世纪 高职高专规划教材

语言 程序设计教程

李淑华 主 编



www.waterpub.com.cn

21 世纪高职高专规划教材

C 语言程序设计教程

李淑华 主 编
张翼英 许薇薇 刘 勇 副主编

中国水利水电出版社

内 容 提 要

本书共 13 章, 内容包括: C 语言的发展和特点; 数据类型、数据的输入和输出语句; 条件语句、循环语句、数组、函数与变量, 以及程序设计方法; 编译预处理; 指针及结构体的概念、程序设计方法; C 语言绘图和文件的有关操作函数等; 最后简单介绍了 C++ 的发展、面向对象程序设计的有关概念。

本书突出应用, 在具有一定深度的同时, 力求通俗易懂, 适合课堂教学。为了检验学生对知识的掌握程度, 本书配备了一定量的复习题和测试题。

本书适合高职高专院校计算机专业的学生学习, 非计算机专业的学生也可以选择性地学习。

本书所配电子教案和源代码可从中国水利水电出版社网站 <http://www.waterpub.com.cn/softdown/> 下载。

图书在版编目 (CIP) 数据

C 语言程序设计教程/李淑华主编. — 北京: 中国水利水电出版社, 2005

21 世纪高职高专规划教材

ISBN 7-5084-2929-X

I. C... II. 李... III. C 语言-程序设计-高等学校: 技术学校-教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 081317 号

书 名	C 语言程序设计教程
作 者	李淑华 主编 张翼英 许薇薇 刘 勇 副主编
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 16 印张 372 千字
版 次	2005 年 8 月第 1 版 2006 年 9 月第 2 次印刷
印 数	5001—8000 册
定 价	22.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前 言

人类伴随时代的跫音进入了 21 世纪, 新世纪将是信息化社会, 信息技术日新月异, 已广泛应用于信息管理、高效处理、辅助设计, 以及人们的日常生活。计算机的应用已成为现代社会生产力发展的重要标志, 信息技术正在不断改变着这个世界。

高等职业教育以培养技术应用型人才为根本任务, 以适应社会需求为目标, 以培养技术应用能力为主线, 旨在提高学生的知识、能力和素质结构。教学内容要突出理论知识的应用和实践能力的培养, 理论课教学要以应用为目的, 专业课教学要加强针对性和实用性。为适应高等职业教育的需要, 针对高职人才培养的特点, 作者编写了这套《C 语言程序设计教程》和《C 语言程序设计教程习题与上机实训指导》教材。

C 语言是一种通用的现代计算机程序设计语言, 它是根据结构化程序设计原则设计并实现的。C 语言不仅适合于系统程序设计, 而且适合于应用程序设计。C 语言在操作系统、工具软件、软件平台、图像处理、数值分析、人工智能和数据库管理系统等方面都得到了广泛的应用。C 语言有很多突出特点: 有较丰富的数据类型, 多种运算符; 语言的组成精练、简洁, 使用方便、灵活, 表达能力强; 同时具有“高级语言”和“低级语言”的双重特点。C 语言提供了某些接近汇编语言的功能, 有利于编写系统软件。

作者写作本书主要出于以下几点考虑:

第一, 在具有一定深度的同时, 力求通俗易懂, 适合课堂教学和读者自学的需要。

第二, 突出应用, 重点介绍了 C 语言最基本、最常用的部分。

第三, 介绍了程序调试方法, 附录中给出了目前常用的 Turbo C 集成开发环境的使用方法。

第四, 全书每章都配有大量的例题、习题, 还配有技能训练。为了检验学生对知识掌握的程度, 本书还配备了一定量的复习题和测试题。

本书不仅适合高职高专计算机专业, 而且还适合非计算机专业, 高级应用和带星号(*) 的习题和技能训练, 对非计算机专业的学生可以不作介绍。

本书由李淑华主编, 张翼英、许薇薇、刘勇任副主编。刘勇编写了第 1 章和第 2 章, 李淑华编写了第 3 章至第 10 章, 张翼英编写了第 11 章和第 13 章, 陈韬编写了第 12 章, 许薇薇编写了各章高级应用和带星号部分的习题及附录部分的内容。参加部分章节编写和大纲讨论的还有姚大鹏、张丕振、栾好利、查玉祥等。

由于水平所限, 书中如有不当之处, 敬请读者批评指正。

编 者

2005 年 6 月

目 录

前言

第1章 C语言概述 1

1.1 C语言发展概述 1

1.2 C语言的特点 1

1.3 简单的C语言程序 2

1.3.1 C语言程序的构成 2

1.3.2 C语言程序的特点 3

1.4 C程序的上机步骤 3

1.4.1 C程序的上机步骤 4

1.4.2 Turbo C系统菜单 4

1.4.3 程序调试方法 6

习题 7

第2章 数据类型与表达式 8

2.1 常量和变量 8

2.1.1 常量 8

2.1.2 变量 8

2.2 整型数据 9

2.2.1 整型常量 9

2.2.2 整型变量 9

2.2.3 整型变量使用 10

2.3 字符型数据 11

2.3.1 字符型常量 11

2.3.2 字符型变量 12

2.3.3 字符串常量 13

2.4 浮点型数据 14

2.4.1 浮点型常量 14

2.4.2 浮点型变量(实型变量) 14

2.5 变量的初始化 14

2.6 运算符及表达式 15

2.6.1 算术运算符和算术表达式 15

2.6.2 赋值语句 16

2.6.3 关系运算符和关系表达式 18

2.6.4 逻辑运算符和逻辑表达式 19

2.6.5 位运算符 19

2.6.6 其他运算符 23

2.7 数据类型转换 24

2.7.1 自动类型转换 24

2.7.2 强制类型转换 25

2.8 类型定义 25

习题 26

第3章 数据的输入和输出 29

3.1 数据的输入 29

3.1.1 字符输入函数 getchar 29

3.1.2 格式输入函数 scanf 29

3.2 数据的输出 31

3.2.1 字符输出函数 putchar 31

3.2.2 格式控制函数 printf 31

3.3 应用举例 33

习题 35

技能训练 36

第4章 条件语句 39

4.1 语句和复合语句 39

4.2 if语句 39

4.2.1 if语句的三种形式 39

4.2.2 if语句的嵌套 42

4.2.3 条件运算符 43

4.3 switch语句 43

4.4 应用举例 46

* 4.5 高级应用 47

习题 50

技能训练 51

第5章 循环控制 54

5.1 goto语句 54

5.2 while循环语句 54

5.3 do_while循环语句 55

5.4 for循环语句 56

5.5 循环的嵌套 57

5.6 几种循环的比较 58

5.7 break 语句和 continue 语句	58	7.7 数组作为函数参数	96
5.7.1 break 语句	58	7.7.1 数组元素作函数实参	96
5.7.2 continue 语句	59	7.7.2 数组名作函数的实参与形参	96
5.8 应用举例	60	7.8 变量的定义位置和作用范围	97
* 5.9 高级应用	62	7.8.1 局部变量	97
习题	65	7.8.2 全局变量	98
技能训练	66	7.9 动态存储变量与静态存储变量	101
第 6 章 数组	67	7.9.1 变量的存储类别	101
6.1 一维数组的定义和引用	67	7.9.2 局部变量的存储方式	102
6.1.1 一维数组的定义	67	7.9.3 全局变量的存储方式	104
6.1.2 一维数组元素的引用	67	7.9.4 存储类别小结	105
6.1.3 一维数组的初始化	68	7.9.5 内部函数和外部函数	106
6.1.4 一维数组应用举例	68	7.10 应用举例	106
6.2 二维数组的定义和引用	70	* 7.11 高级应用	108
6.2.1 二维数组的定义	70	习题	110
6.2.2 二维数组的引用	71	技能训练	111
6.2.3 二维数组的初始化	71	第 8 章 编译预处理	112
6.2.4 二维数组应用举例	72	8.1 宏定义	112
6.3 字符数组与字符串	74	8.1.1 不带参数的宏定义	112
6.3.1 字符数组	74	8.1.2 带参数的宏定义	113
6.3.2 字符串	76	8.2 “文件包含”处理	113
6.4 应用举例	80	8.3 条件编译	114
* 6.5 高级应用	81	8.3.1 # ifdef 语句	114
习题	85	8.3.2 # ifndef 语句	114
技能训练	86	8.3.3 # if 语句	114
第 7 章 函数与变量	88	8.3.4 # undef 语句	115
7.1 函数的概念	88	8.4 应用举例	115
7.2 函数的定义	89	习题	116
7.2.1 无参函数的定义	89	技能训练	116
7.2.2 有参函数的定义	89	第 9 章 指针	118
7.3 函数参数和函数值	90	9.1 指针的定义与引用	118
7.3.1 形式参数和实际参数	90	9.1.1 指针的概念	118
7.3.2 函数的返回值	91	9.1.2 指针变量的定义	118
7.4 函数的调用	91	9.1.3 指针变量的引用	119
7.4.1 函数调用的一般形式	91	9.2 指针变量作为函数参数	120
7.4.2 函数调用的方式	92	9.3 指针与数组	121
7.4.3 对被调用函数的说明	92	9.3.1 指向数组元素的指针变量	
7.5 函数的嵌套调用	93	的定义与赋值	121
7.6 函数的递归调用	94	9.3.2 通过指针引用数组元素	122

9.3.3 指针变量的运算	124	10.7.3 链表首指针和结尾标志	144
9.3.4 数组名作函数参数	125	10.8 共用体	148
9.4 指针与字符串	127	10.8.1 “共用体”类型变量的定义	149
9.4.1 字符数组	127	10.8.2 共用体变量的引用方式	149
9.4.2 字符指针	127	10.8.3 共用体类型数据的特点	149
9.5 指针数组	128	10.9 枚举类型	151
9.5.1 指针数组的引用	128	* 10.10 高级应用	153
9.5.2 命令行参数	129	习题	154
* 9.6 高级应用	130	技能训练	156
习题	132	第 11 章 C 语言绘图	157
技能训练	133	11.1 微机的显示系统	157
第 10 章 结构体与共用体	134	11.2 Turbo C 图形函数	157
10.1 结构体的定义	134	11.2.1 图形控制函数	158
10.2 定义结构体类型的变量	135	11.2.2 颜色控制函数	160
10.2.1 先定义结构体类型 再定义变量名	135	11.2.3 基本图形函数	161
10.2.2 在定义类型的同时定义变量	135	11.3 应用举例	166
10.2.3 直接定义结构类型变量	136	习题	169
10.3 结构体类型变量的引用	137	技能训练	170
10.3.1 结构体类型变量的引用格式	137	第 12 章 文件	171
10.3.2 结构体变量的引用规则	137	12.1 文件概述	171
10.4 结构体变量的初始化	138	12.1.1 文件的存储方式	171
10.4.1 对外部存储类型的结构体 变量进行初始化	138	12.1.2 C 的文件操作	171
10.4.2 对静态存储类型的结构体 变量进行初始化	139	12.1.3 缓冲文件与非缓冲文件	171
10.5 结构体数组	139	12.1.4 文件类型指针	172
10.5.1 结构体数组的定义	139	12.2 文件的打开与关闭	173
10.5.2 结构体数组的初始化	140	12.2.1 文件的打开(fopen 函数)	173
10.5.3 举例	140	12.2.2 文件的关闭(fclose 函数)	174
10.6 指向结构体类型数据的指针	141	12.3 文件的读写	175
10.6.1 指向结构体变量的指针	141	12.3.1 读写一个字符	175
10.6.2 指向结构体数组的指针	142	12.3.2 读写一个字符串	177
10.7 用指针处理链表	143	12.3.3 读写数据字段	178
10.7.1 链表	143	12.3.4 文件的格式化读写	179
10.7.2 建立链表的方法——结构体	143	12.4 文件定位与随机读写	180
		12.5 文件的出错检测	181
		12.6 应用举例	182
		* 12.7 高级应用	185
		习题	185
		技能训练	186

第 13 章 C++ 简介	187	13.5 多态性	198
13.1 C++ 与面向对象程序设计	187	习题	198
13.1.1 面向对象程序设计	187	附录 A Turbo C 集成开发环境	199
13.1.2 Windows 平台上的 C++ 程序 开发工具	188	附录 B 运算符的优先级及其结合性 ...	204
13.1.3 面向对象的主要概念	188	附录 C 标准 ASCII 字符集	205
13.2 类说明	189	附录 D Turbo C 2.0 常用库函数	207
13.3 对象说明	191	附录 E 复习题	213
13.4 继承	195	附录 F 测试题	236

第 1 章 C 语言概述

C 语言是计算机高级程序设计语言之一。它设计精巧，功能齐全，既适合编写应用软件，又适合作为系统描述语言编写系统软件。

1.1 C 语言发展概述

C 语言是于 1972 年至 1973 年间由美国贝尔实验室的 M. D. Ritchie 和 K. Thompson 以及英国剑桥大学的 M. Richards 等为描述和实现 UNIX 操作系统而设计的。UNIX 操作系统源代码的 90% 以上是用 C 语言编写的。UNIX 操作系统所具有的一些主要特点，如便于理解，易于修改，具有良好的可移植性等，在一定程度上都受益于 C 语言，所以 UNIX 操作系统的成功与 C 语言是密不可分的。

最初的 C 语言是附属于 UNIX 的操作系统环境，而它的产生又可以更好地描述 UNIX 操作系统。时至今日，C 语言已独立于 UNIX 操作系统，成为微型、小型、中型、大型和超大型（巨型）计算机上通用的一种程序设计语言。M. D. Ritchie 和 K. Thompson 也以他们在 C 语言和 UNIX 系统方面的卓越贡献获得了很高的荣誉。1982 年，他们获得了美国《电子学》杂志颁发的成就奖，成为该奖自颁发以来首次因软件工程成就而获此殊荣的获奖者。1983 年，他们又获得了计算机界的最高荣誉奖——图灵奖。

随着 C 语言的不断发展、应用和普及，目前，C 语言已经能够在多种操作系统如 UNIX、DOS 等环境下运行，实用的 C 语言编译系统也种类繁多，如 Microsoft C、Turbo C 等。C 语言是近年来最受人们欢迎的系统程序设计语言之一。

1.2 C 语言的特点

C 语言能够成为目前应用广泛的高级程序设计语言之一，完全是由其语言特点决定的。C 语言的特点可大致归纳如下：

(1) 基本语言组成部分紧凑简洁。C 语言只有 32 个标准关键字，44 个标准运算符以及 9 种控制语句，不但语言的组成精练、简洁，而且使用方便、灵活。

(2) C 语言运算符丰富，表达能力强。C 语言兼具“高级语言”和“低级语言”的双重特点，其运算符包含的内容广泛，所生成的表达式简练、灵活，有利于提高编译效率和目标代码的质量。

(3) C 语言数据结构丰富，结构化好。C 语言提供了编写结构化程序所必需的各种数据结构和控制结构，这些丰富的数据和控制结构以及函数调用为主的程序设计风格，保证了利用 C 语言编写的程序能够具有良好的结构化。

(4) 具有结构化的控制语句。如 if... else 语句、while 语句、do... while 语句、switch

语句、for 语句等。用函数作为程序模块以实现程序的模块化，是结构化的理想语言，符合现代编程风格。

(5) C 语言提供了某些接近汇编语言的功能，有利于编写系统软件。C 语言提供的一些运算和操作，能够实现汇编语言的一些功能，如它可以直接访问物理地址，并能进行二进制位运算等，这都为编写系统软件提供了方便条件。

(6) C 语言程序所生成的目标代码质量高。用 C 语言程序所生成的目标代码效率仅比用汇编语言描述同一个问题低 20% 左右。

(7) C 语言程序可移植性好。C 语言所提供的语句中，没有直接依赖于硬件的语句，与硬件有关的操作，如数据的输入、输出等都是通过调用系统的库函数来实现的，而库函数本身不是 C 语言的组成部分。因此用 C 语言编写的程序可以很容易地从一种计算机环境移植到另一种计算机环境中。

C 语言的弱点：一是运算符的优先级较复杂，不易记忆；二是由于 C 语言的语法限制不太严格，增强程序设计灵活性的同时，在一定程度上也降低了某些安全性，因此对程序设计人员提出了更高的要求。

1.3 简单的 C 语言程序

任何一种程序设计语言，都具有特定的语法规则和一定的表示形式。程序的书写格式和程序的构成规则是程序设计语言表示形式的一个重要方面。按照一定的格式和构成规则书写程序，不仅可以使程序设计人员和使用程序的人员易理解，更重要的是，把程序输入到计算机时，计算机能够充分识别，并能正确地执行它。

1.3.1 C 语言程序的构成

[例 1.1] 编写一个 C 语言程序，显示一个字符串 “Good morning, class!”。

```
/* file name a1.c */
main ()
{
    printf (" Good morning, class! \n");
}
```

编译 a1.c 程序，并执行文件 a1.exe，执行结果为：

C: \>a1 ↙

Good morning, class!

[例 1.2] 将两个整数之和显示出来。

```
/* file name a2.c */
main ()
{
    int a, b, sum;                /* 定义 a, b, sum 三个整型变量 */
    a= 100; b= 200;               /* 给变量 a 和 b 赋值 */
    sum= a+ b;                    /* 计算 a 与 b 的和，存入变量 sum 中 */
}
```

```
printf (" sum= % d\n", sum);    /* 输出 sum 的值 */  
}
```

编译 a2.c 程序, 并执行文件 a2.exe, 执行结果为:

C: \>a2 ✓

sum=300

[例 1.3] 编写计算长为 a, 宽为 b 的长方形面积 s 的 C 程序。

```
/* file name a3.c */  
#include "stdio.h"  
main ()  
{  
    int a, b, s;                /* 定义 a, b, s 三个整型变量 */  
    printf (" a, b= ?");  
    scanf ("% d,% d", &a, &b);  /* 读入两个整数, 存入变量 a 和 b 中 */  
    s= a* b;                    /* 计算长方形的面积 s 的值 */  
    printf (" s= % d\n", s);    /* 输出面积 s 的值 */  
}
```

编译 a3.c 程序, 并执行文件 a3.exe, 执行结果为:

C: \>a3 ✓

a, b=? 20, 10 ✓

s=200

1.3.2 C 语言程序的特点

- (1) C 程序是由函数构成的, 其中至少包括一个函数, 即主函数 main ()。
- (2) 函数体是由左右花括号 { } 括起来的部分。
- (3) C 语言中的每条基本语句都以 “;” 结束。
- (4) C 语言书写格式自由, 一行内可以写一条语句, 也可以写多条语句。
- (5) #include 语句是编译处理语句, 其作用是将由双引号或尖括号括起来的文件内容读入该语句位置处。输入输出库函数一般需要使用 #include 语句将 "stdio.h" 文件包含到源文件中。#include 语句的使用方法将在第 8 章介绍。

对程序的说明:

- (1) 可用 /* ... */ 对 C 程序中的任何部分作注释。
- (2) C 语言中所有变量都必须先定义类型, 然后使用。
- (3) 一个 C 语言程序通过函数之间的相互调用来实现相应的功能。既可以是系统提供的库函数, 也可以是根据需要自己定义的函数。

1.4 C 程序的上机步骤

C 语言程序上机运行, 需要一个运行环境, 常用的运行环境有: UNIX 操作系统下、DOS 操作系统下的 MS 4.0 和 Turbo C。其中, 使用最方便的是 Turbo C, 因此本书以

Turbo C 来讲解其使用方法。

1.4.1 C 程序的上机步骤

1. 编辑

在 Turbo C 环境下，将 C 语言源程序通过键盘输入到计算机中，并以文件形式存盘，源程序都是以 .C 为扩展名。

2. 编译

通过编辑程序将源程序输入到计算机后，需要经过 C 语言编译器将其生成目标程序。得到的目标文件都是以 .OBJ 为文件扩展名。若源程序有错误，需要对其进行修改，然后重新编译。

3. 连接

经过编译后生成的目标文件不能直接执行，需要经过连接之后才能生成可执行代码。在 DOS 系统下，连接后所得到的可执行文件扩展名都是 .EXE。

所谓连接，是将目标文件和库函数或其他目标程序连接成可执行的目标程序。

4. 执行

连接后源程序就可以生成可执行文件了。在 DOS 系统下，只要键入可执行文件名，按回车键就可以了。

综上所述，上机步骤为：编辑 (.C) → 编译 (.OBJ) → 连接 (.EXE) → 执行，如图 1.1 所示。

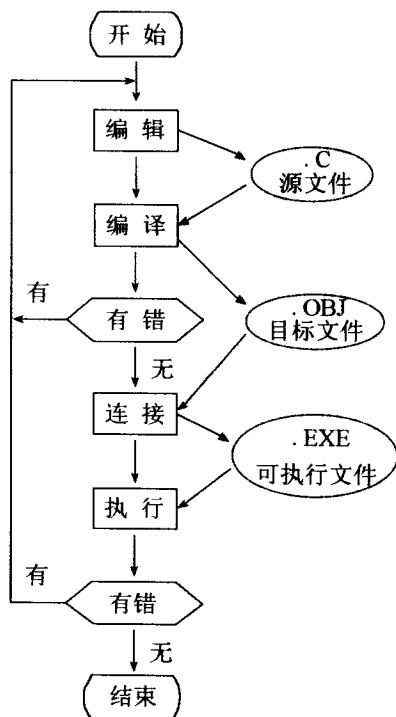


图 1.1 C 语言的上机步骤

1.4.2 Turbo C 系统菜单

Turbo C 不仅是一个快速、高效的编译软件，同时还带有一个易学、好用的集成环境。使用 Turbo C 时无须独立的编辑、编译和连接程序，就能编辑和运行 C 程序，所有这些功能都组合在 Turbo C 的集成环境内，并且可以通过菜单选择使用这些功能。

1. Turbo C 2.0 集成开发环境的主屏幕

Turbo C 2.0 系统是 DOS 环境下的应用软件。只要在 DOS 系统下进入 Turbo C 2.0 所在的目录，输入 TC，回车后（安装 Turbo C 2.0 时的默认设置），即进入 Turbo C 2.0 集成环境。按任意键，版本信息消失，屏幕显示如图 1.2 所示。

Turbo C 2.0 集成开发环境的主屏幕，由上至下分成四部分：主菜单，编辑窗，信息窗和默认的快速参考行。

File Edit Run Compile Project Options Debug Break/Watch

Line	1	Col	1	Insert	Indent	Tab Fill	Unindent	C: NONAME.C
Message								
F1—Help F5—Zoom F6—Switch F7—Trace F8—Step F9—Make F10—Menu								

图 1.2 Turbo C 2.0 集成开发环境主屏幕

2. 主菜单功能

Turbo C 2.0 的集成开发环境的主菜单一共有 8 个选择项，其功能简介如下：

File	装入或保存文件，管理目录，转入 DOS 和退出 Turbo C
Edit	建立和编辑文件
Run	编译、连接和运行装入环境下的当前程序
Compile	编译在环境下的当前程序
Project	管理多文件工具
Options	设置编译程序和连接程序的各种选择项
Debug	设置各种调试选择项
Break/Watch	允许用户增加、删除和编辑监视表达式，可设置和清除断点

3. 文件菜单及功能

File 菜单如图 1.3 所示，共有 9 个子菜单，其中：

Load	F3
Pick	Alt+F3
New	
Save	F2
Write to	
Directory	F4
Change dir	
Os shell	
Quit	Alt+X

图 1.3 File 菜单

- Load 装入一个新文件，此命令的热键是 F3。
- Pick 在文件列表中选择一个或几个不同的文件装入编辑窗口。热键是 Alt+F3。
- New 选择此项将建立一个新文件，此文件名暂时是 NONAME.C。
- Save 把编辑窗口中的文件存盘。
- Write to 将文件写入新的文件名下或覆盖一个已存在的文件，若覆盖，屏幕有提示。
- Directory 显示当前目录中的文件列表。按 F4 键，可以选择不同驱动器和不同目录。
- Change dir 显示当前目录或选择不同的驱动器和目录。

Os shell 暂时退出 Turbo C, 返回 DOS 状态, 输入 EXIT 并按回车键, 返回 Turbo C。

Quit 退出 Turbo C, 返回 DOS 状态下。

有关 Turbo C 2.0 的详细操作, 请见附录 2。

1.4.3 程序调试方法

一次编写一个程序难免有错误, 一个计算机程序必须对其进行测试以确定其是否可以正确工作, 这一过程称为调试。程序错误有两种: 一种是语法错误, 另一种是运行错误。语法错误是由于指令没有按照程序设计语言的语法规则编写所致, 一般通过编译和连接就能找到, 可根据提示修改错误, 直到通过编译和连接为止, 这时就说明程序没有语法错误。当程序已顺利执行, 但得到的却不是所期望的结果, 说明程序存在运行错误。这类错误除由类型错误造成的外都归为逻辑错误。计算机不能指出运行错误, 只能由编程者自己查找出程序的错误所在。运行错误往往是由于编程者所采用的变量类型或(部分或全部)算法有问题, 而导致开始或中间结果不正确。这就要求有能快速找出错误的方法, Turbo C 集成环境提供了方便灵活的手段控制程序的执行, 可在任意语句行上暂时或永久停止程序的执行(程序停止执行的语句行称为断点), 并可追踪、观察中间(变量)结果。下面介绍一些调试手段。

1. 设置和使用观察变量

找出程序中的运行错误的一种有效方法是能知道程序执行过程中某些变量或表达式值的变化, TC (Turbo C) 集成环境提供了观察变量的途径。选 Break/watch 菜单项中的 Add watch 项或同时按 Ctrl 和 F7 键, 会出现一个输入框, 输入欲观察变量名或表达式后, 它们就会出现在 watch (屏幕底部) 窗口中。重复使用 Break/watch 菜单项中的 Add watch 项或同时按 Ctrl 和 F7 键可以在 watch 窗口添加多个变量或表达式。

变量或表达式设置完成后, 执行程序。watch 窗口中变量或表达式的值随程序的执行可能发生改变, 由于程序执行速度非常快, 人眼难以看清这种变化。因此, 需要有能控制程序执行的方法, 以有足够的时间观察变量或表达式的值。下面介绍的单步执行程序、设置和使用断点提供了观察变量或表达式值的方法。

2. 单步执行程序

选 Run 菜单中的 Step over 命令项或按 F8 键, 亮条就出现在主函数处, 说明程序从主函数开始执行。每次选 Run 菜单中的 Step over 命令项或按一次 F8 键则执行一行程序, 遇到函数时不进入函数(定义), 即函数作为一条语句或一个表达式处理。

Run 菜单中的 Trace into 命令项或按 F7 键也是单步执行程序, 不同的是遇到函数时进入其内部逐行执行。

Run 菜单中的 Goto cursor 命令项或按 F4 键, 是从主函数开始执行程序到光标所在处(跟踪到光标)。欲让程序在某条语句暂停, 只需将光标移到那条语句所在行, 选 Run 菜单中的 Goto cursor 命令项或按 F4 键, 程序即可暂停在光标所在行。然后再移动光标到欲暂停的语句行上, 再选 Run 菜单中的 Goto cursor 命令项或按 F4 键, 由此可以方便地调试程序。

3. 设置和使用断点

通过设置断点, 可以分段解决问题, 把出现问题的范围缩小。调试程序时, 可以在程

序中设置多个断点。当程序执行到断点时，就会在有断点的那一行上暂停程序的执行，显示所需观察变量或表达式的当前值，用户据此判定断点前的程序是否有问题，并将调试控制返回给用户，即用户可以接着连续执行、单步执行、跟踪到光标及断点执行。

设置断点的步骤为：

(1) 在编辑窗口将光标移到欲设置断点的那一行上。

(2) 选 Break/watch 菜单项中的 Toggle breakpoint 命令项或同时按 Ctrl 和 F8 键，被设置断点的一行呈高亮度显示。

(3) 欲设多个断点，重复步骤 (1)、(2)，断点所在的行均呈高亮度显示。

(4) 要删除断点，将光标移到欲删除断点所在行，选 Break/watch 菜单项中的 Toggle breakpoint 命令项或同时按 Ctrl 和 F8 键。

以上方法可结合使用。程序连续执行时，一遇到断点就会暂时停止，并返回编辑状态。断点不会因为被执行而清除。

习 题

1.1 概述 C 语言的主要特点及用途。

1.2 构成 C 语言程序的基本单位是什么？它由哪几部分组成？

1.3 简述 C 语言设计和执行程序的全过程。

1.4 编写输出以下信息的程序。

```
#####  
Turbo C 2.0  
#####
```

1.5 上机运行本章的 3 个例题。

1.6 写出下面程序的运行结果，并上机调试。

```
main ()  
{int a1, a2, x;  
  a1= 100; a2= 50; x= a1- a2;  
  printf (" x= % d\n", x);  
}
```

1.7 写出下面程序的运行结果，并上机调试。

```
main ()  
{int a1, a2, x;  
  a1= 10; a2= 20; x= a1* a2;  
  printf (" a1= % d, a2= ,% d\n", a1, a2);  
  printf (" x= % d\n", x);  
}
```

第 2 章 数据类型与表达式

任何程序都涉及到待处理的数据，数据可以是常量和变量。本章将讨论 C 语言的变量、常量、基本数据类型、运算符和表达式，以及不同数据类型之间的转换等。

2.1 常量和变量

2.1.1 常量

常量又称常数，是指在程序运行中其值不能被改变的量。常量可分为不同的类型，可以用一个名字（字符序列）来代表一个常量，这种常量称为符号常量。符号常量的使用将在第 8 章介绍。

2.1.2 变量

变量是指在程序运行过程中其值可以被改变的量。变量被分为不同类型，在内存中占用不同的存储单元，以便用来存放相应变量的值。

用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符，标识符就是一个名字。

1. 标识符的有关规定

(1) 由英文字母、数字、下划线组成，且第一个字符必须是字母或下划线，一般不超过 8 个字符。

(2) 大小写字母的含义不同。

(3) 不能使用 C 语言中的关键字做标识符，如 `int`。

(4) 用户取名时，应当尽量遵循“简洁明了”和见名知意的原则。

2. 变量名的有关规定

(1) 使用变量前，一定先定义，后使用，否则不能作为变量名。

`int student; statent=20;` 在编译时检查出 `statent` 未经定义。

(2) 每一个变量属于一种类型，以便在编译时据此检查该变量所进行的运算是否合法。

合法的变量名：`day`, `year`, `book`, `student`, `a1`, `a2`, `b1`, `b2`。

不合法的变量名：`#852`, `. com`, `$s1`, `1999y`, `123`, `float`。

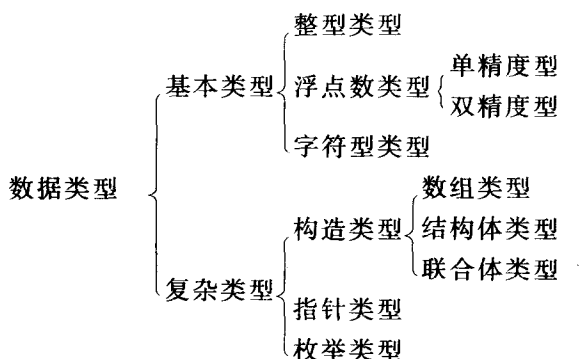
3. 变量名的定义

[存储属性]	〈数据类型〉	〈变量名表〉
--------	--------	--------

变量定义包括三部分：存储属性、数据类型、变量名表。存储属性决定了变量的存在性和可见性，这将在第 7 章介绍。数据类型决定了变量的取值范围和占用内存空间的字节

数，变量名表是具有同一数据类型变量的集合。

4. 数据类型



2.2 整型数据

2.2.1 整型常量

(1) 十进制整数：以数字直接开头的是十进制数。如 1236、-234、0。

(2) 八进制整数：以 0 开头的常量是八进制数。如 0123 表示八进制数的 $(123)_8$ ，-011 表示八进制数的 $(-11)_8$ 。

(3) 十六进制整数：以 0x 开头的数是十六进制数。如 0x123，即 $(123)_{16}$ 。

[例 2.1] 三种进制表示方法的转换。

源程序 a1.c

```

main ()
{
    int x= 1246, y= 01246, z= 0x1246;
    printf ("% d,% d,% d\n", x, y, z);      /* % d: 以十进制格式符输出 */
    printf ("% o,% o,% o\n", x, y, z);      /* % o: 以八进制格式符输出 */
    printf ("% x,% x,% x\n", x, y, z);      /* % x: 以十六进制格式符输出 */
}

```

C: \>a1 ✓

1246, 678, 4678

2336, 1246, 11106

4de, 2a6, 1246

2.2.2 整型变量

C 语言规定使用变量之前，必须“先定义，后使用”。整型变量可用来存放整型数据（即不带小数点的数）。整型变量可以分为：基本型、短整型、长整型和无符号型 4 种类型，其定义的关键字如下。

(1) 基本型：用 int 表示。