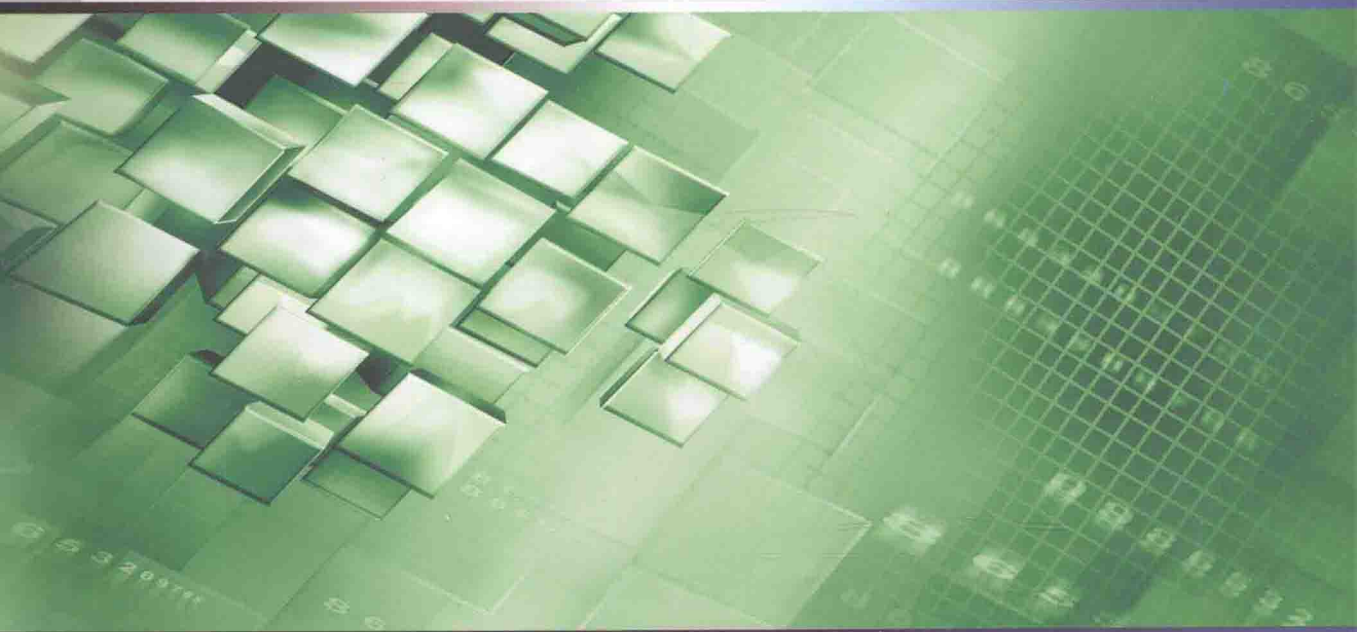




全国普通高等教育“十二五”规划教材

C语言程序设计教程

主 编 杨 旭 李 杰
主 审 张 华



时代出版传媒股份有限公司
北京时代华文书局

全国普通高等教育“十二五”规划教材

C 语言程序设计教程

主 编 杨 旭 李 杰

主 审 张 华



ARPTIME
时代出版

时代出版传媒股份有限公司
北京时代华文书局

图书在版编目 (CIP) 数据

C 语言程序设计教程 / 杨旭, 李杰主编. -- 北京 :
北京时代华文书局, 2014.1

ISBN 978-7-80769-216-4

I. ①C… II. ①杨… ②李… III. ①C 语言—程序设
计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 305148 号

C 语言程序设计教程

杨旭 李杰/主编

出 版 人 田海明 周殿富

责任编辑 胡俊生 特约编辑 章银武

出 版 时代出版传媒股份有限公司 <http://www.press-mart.com>

北 京 时 代 华 文 书 局 <http://www.bjsdsj.com.cn>

北京东城区安定门外大街 136 号皇城国际大厦 A 座 8 楼 邮编 100011

发 行 北京时代华文书局图书发行部 (010) 64267120 64267397

印 刷 北京长阳汇文印刷厂印制 (010) 80356121

规 格 787mm×1092mm 1/16

字 数 387 千字

印 张 15.5

版 次 2014 年 3 月第 1 版 2014 年 3 月第 1 次印刷

书 号 ISBN: 978-7-80769-216-4

定 价 35.00 元

※ 如发现因印装质量影响阅读, 请与印刷厂联系调换 ※ 版权所有 侵权必究

前 言

C 语言是一种计算机程序设计语言，它既具有高级语言的特点，又具有低级语言的特点；它既可以编写系统软件，也可以用来编写应用软件。C 语言的应用范围非常广泛，它具有很强的数据处理能力，不仅计算功能、逻辑判断功能比较强大，图形功能也很强大，适合于 DOS、UNIX、Windows、Linux 等多种操作系统。

► 本书特点

为了帮助广大学生更好地掌握 C 语言编程技术，帮助教师更好地组织教学，我们特组织专家和一些一线骨干教师编写了《C 语言程序设计教程》一书。本书具有如下几个特点。

(1) 本书是一本全面介绍 C 语言程序设计的图书，改变传统教材以语法为驱动的内容编排模式，有利于任课教师组织课堂教学，有利于培养学生解决实际问题的能力。

(2) 本书注重教材的可读性和实用性，每章开头都给出了本章概览、本章重点和本章目标，对关键知识点进行了详细的说明，并附有大量的图表方便读者正确、直观地对问题进行理解，样例程序由浅入深，强化了知识点、算法、编程方法与技巧，并给出了详细的解释。

(3) 本书以实用为出发点，以培养读者的实践和实际应用能力为目标，并通过通俗易懂的文字和手把手的教学方式讲解 C 语言中的要点、难点，使读者不仅能掌握程序设计知识，还能掌握实际的应用技能。

(4) 本书每章后面都有本章小结和大量的思考与练习，以让读者及时巩固所学的知识。

(5) 本书内容先进，体系合理，概念清晰，讲解详尽，分散难点，例题丰富，深入浅出，文字流畅，通俗易懂，是学习 C 程序设计的理想教材。

► 本书结构安排

本书结构安排如下：

第 1 章 C 语言基本知识。通过本章学习，读者可以：了解计算机编程语言和语言的特点；掌握 C 程序的基本组成；了解并配置 Turbo C 2.0 的集成环境，对源程序进行编译、链接并运行。

第 2 章 数据类型、运算符与表达式。通过本章学习，读者可以：了解 C 语言中的数据类型、常量与变量的区别；掌握整型、实型、字符型常量和变量的声明和使用方法；了解变量的初始化方法；掌握不同类型数据的相互转换、运算符和表达式。

第 3 章 C 程序的结构化程序设计。通过本章学习，读者可以：了解 C 语言中语句的类型和特点；掌握常用语句的功能与语法；掌握选择、循环、跳转语句的功能与语法。

第 4 章 利用数组处理批量数据。通过本章学习，读者可以：了解一维、二维和

字符数组的定义；掌握一维、二维的引用和初始化方法；了解字符串和字符串结束的标志、用于字符串处理的几个函数。

第5章 用函数实现模块化程序设计。通过本章学习，读者可以：了解函数的定义和声明方法、函数的形参、实参和函数值的意义和使用方法；掌握函数的三种调用方式、使用数组作为函数参数的方法、局部变量和全局变量的声明和使用方法、在不同程序间使用全局变量的方法；理解动态变量、外部变量使用方法；掌握内部函数和外部函数的声明和使用方法。

第6章 善于利用指针。通过本章学习，读者可以：了解指针的概念；掌握变量指针、数组指针、字符串指针和函数指针的用法；掌握指针数组的用法。

第7章 建立自己的数据类型。通过本章学习，读者可以：了解结构类型的一般形式和定义；掌握结构变量的引用和初始化，了解结构数组和指向结构的指针；掌握引用自身的结构以及链表操作；掌握联合和枚举的定义、说明，以及其变量的赋值和使用方法。

第8章 对文件的输入输出。通过本章学习，读者可以：了解文件的类型及特点；掌握文件的打开、关闭、读写和定位方法；掌握文件读写出错的检测方法。

第9章 编译预处理命令。通过本章学习，读者可以：掌握宏定义方法、文件包含方法和条件编译方法。

第10章 位运算。通过本章学习，读者可以：了解位运算的定义；掌握各种位运算的功能及相应的位运算符；掌握位域的概念、声明和使用方法。

➤ 本书创作人员

本书由杨旭、李杰任主编，周铮、钱志林任副主编，张华任主审。其中第2、3、5、10章由杨旭负责编写，第1、4、7章由李杰负责编写，第6、8由周铮负责编写，第9章和附录由钱志林负责编写。

➤ 本书适合对象

本书为全国普通高等教育“十二五”规划教材，适合计算机专业的本、专科生使用，也可以作为大学各专业公共教材和全国计算机等级考试参考书。

本书在编写过程中，难免有疏漏和不当之处，敬请各位专家及读者不吝赐教。

编 者
2013年12月

目 录

第 1 章 C 语言基本知识	1	2.4.5 条件运算符	27
【本章概览】	1	2.4.6 关系运算符	28
【本章重点】	1	2.4.7 逻辑运算符	28
【本章目标】	1	2.4.8 位运算符	29
1.1 C 语言概述	1	2.5 本章小结	32
1.1.1 计算机编程语言	1	2.6 思考与练习	32
1.1.2 C 语言的发展历史	2	一、填空题	32
1.1.3 C 语言的特点	2	二、选择题	33
1.2 C 语言程序的结构	3	三、程序分析题	35
1.3 运行 C 程序的步骤与方法	5	四、改错题	36
1.3.1 进入 Turbo C 2.0 集成环境	5	五、简答题	37
1.3.2 配置运行环境	6	第 3 章 C 程序的结构化程序设计	38
1.3.3 编辑源程序	7	【本章概览】	38
1.3.4 对源程序进行编译、链接并运行	7	【本章重点】	38
1.3.5 查看运行结果	7	【本章目标】	38
1.3.6 退出 Turbo C 2.0 集成环境	7	3.1 C 程序的语句	38
1.4 本章小结	8	3.1.1 表达式语句	39
1.5 思考与练习	8	3.1.2 函数调用语句	39
一、填空题	8	3.1.3 控制语句	39
二、操作题	8	3.1.4 复合语句	40
第 2 章 数据类型、运算符与表达式	9	3.1.5 空语句	40
【本章概览】	9	3.2 顺序结构	40
【本章重点】	9	3.2.1 赋值语句	40
【本章目标】	9	3.2.2 字符数据的输入与输出函数	41
2.1 C 语言的数据类型	9	3.2.3 格式输入和输出函数	44
2.2 C 语言的标识符	10	3.3 选择结构	49
2.3 常量与变量	11	3.2.1 关系运算与逻辑运算	49
2.3.1 常量	11	3.2.2 if 语句	51
2.3.2 变量	15	3.2.3 switch 语句	56
2.3.4 不同类型数据的相互转换	23	3.4 循环结构	59
2.4 C 语言的运算符与表达式	24	3.3.1 goto 语句	59
2.4.1 算术运算符	24	3.3.2 while 语句	60
2.4.2 自增自减运算	25	3.3.3 do_while 语句	61
2.4.3 逗号运算符	26	3.3.4 for 语句	62
2.4.4 赋值运算符	26	3.3.5 循环语句的嵌套	65



3.3.6 break 语句和 continue 语句.....	66	5.1.2 函数的声明.....	103
3.5 本章小结.....	67	5.1.3 函数的定义、声明和调用的区别.....	104
3.6 思考与练习.....	67	5.2 函数的参数和函数的值.....	105
一、填空题.....	67	5.2.1 形参和实参.....	105
二、选择题.....	70	5.2.2 函数的值.....	106
三、简答题.....	74	5.3 函数的调用.....	107
四、编程题.....	74	5.3.1 函数调用的方式.....	107
第 4 章 利用数组处理批量数据.....	76	5.3.2 嵌套调用.....	108
【本章概览】.....	76	5.3.3 递归调用.....	110
【本章重点】.....	76	5.4 数组作函数参数.....	113
【本章目标】.....	76	5.4.1 数组元素作为函数参数.....	113
4.1 一维数组.....	76	5.4.2 数组名作为函数的实参.....	115
4.1.1 一维数组的定义.....	76	5.1 局部变量和全局变量.....	117
4.1.2 一维数组元素的引用.....	77	5.1.1 局部变量.....	117
4.1.3 一维数组的初始化.....	79	5.1.2 全局变量.....	119
4.1.4 一维数组的实例.....	81	5.2 变量的存储类型.....	123
4.2 二维数组.....	83	5.2.1 动态变量.....	123
4.2.1 二维数组的定义.....	83	5.2.2 静态变量.....	125
4.2.2 二维数组元素的引用.....	84	5.2.3 外部变量.....	126
4.2.3 二维数组的初始化.....	85	5.3 内部函数和外部函数.....	127
4.2.4 二维数组的实例.....	86	5.3.1 内部函数（静态函数）.....	127
4.3 字符数组.....	89	5.3.2 外部函数.....	127
4.3.1 字符数组的定义.....	89	5.5 本章小结.....	128
4.3.2 字符数组的初始化.....	89	5.6 思考与练习.....	129
4.3.3 字符串和字符串结束标志.....	90	一、填空题.....	129
4.3.4 字符数组的引用和输入输出.....	93	二、选择题.....	131
4.3.5 用于字符串处理的几个函数.....	93	三、编程题.....	133
4.3.6 字符数组的实例.....	96	第 6 章 善于利用指针.....	134
4.4 本章小结.....	97	【本章概览】.....	134
4.5 思考与练习.....	97	【本章重点】.....	134
一、填空题.....	97	【本章目标】.....	134
二、选择题.....	98	6.1 变量的地址和指针变量.....	134
三、编程题.....	99	6.1.1 变量的地址.....	134
第 5 章 用函数实现模块化程序设计.....	100	6.1.2 指针变量.....	135
【本章概览】.....	100	6.1.3 指针变量的使用.....	135
【本章重点】.....	100	6.2 数组指针.....	137
【本章目标】.....	100	6.2.1 数组指针的运算.....	137
5.1 函数的基本知识.....	100	6.2.2 数组元素的指针.....	138
5.1.1 函数定义的一般形式.....	101	6.2.3 数组名作函数实参.....	140



6.2.4 指向二维数组元素的指针变量	144	7.5 本章小结	179
6.2.5 指向一维数组的指针变量	145	7.6 思考与练习	180
6.3 字符串指针	146	一、填空题	180
6.3.1 字符串的表示形式	146	二、编程题	181
6.3.2 字符串指针作函数参数	148	第 8 章 对文件的输入输出	182
6.4 函数指针	149	【本章概览】	182
6.4.1 指向函数的指针变量	149	【本章重点】	182
6.4.2 返回指针值的函数	150	【本章目标】	182
6.5 指针数组	151	8.1 文件的基本知识	182
6.5.1 指针数组的定义和使用	151	8.1.1 文件的类型	182
6.5.2 命令行参数	152	8.1.2 文件指针	183
6.6 二级指针	153	8.1.3 流 (stream) 和文件 (file)	184
6.7 本章小结	154	8.1.4 关于 C 的库文件	184
6.8 思考与练习	154	8.2 文件的打开与关闭	184
一、写出以下程序的运行结果	154	8.2.1 文件的打开 (fopen ()) 函数	185
二、编程题	157	8.2.2 文件的关闭 (fclose ()) 函数	186
第 7 章 建立自己的数据类型	158	8.3 文件的操作	187
【本章概览】	158	8.3.1 字符读写函数 (fputc 和 fgetc)	187
【本章重点】	158	8.3.2 字符串读写函数 (fputs 和 fgets)	190
【本章目标】	158	8.3.3 数据块读写函数 (fread 和 fwrite)	192
7.1 结构类型	158	8.3.4 格式化读写 (fscanf 函数和 fprintf 函数)	194
7.1.1 结构类型的一般形式	158	8.3.5 文件的定位函数	195
7.1.2 结构变量的定义	159	8.3.6 文件检测函数	196
7.1.2 结构变量的引用	161	8.4 典型程序实例	197
7.1.3 结构变量的初始化	162	8.5 本章小结	198
7.1.4 结构数组	163	8.6 思考与练习	199
7.1.5 指向结构的指针	165	一、填空题	199
7.1.6 结构指针变量作函数参数	168	二、编程题	200
7.1.7 引用自身的结构	169	第 9 章 编译预处理命令	201
7.1.8 链表操作	171	【本章概览】	201
7.2 联合	174	【本章重点】	201
7.2.1 联合的定义	174	【本章目标】	201
7.2.2 联合变量的说明	175	9.1 宏定义	201
7.2.3 联合变量的赋值和使用	176	9.1.1 不带参数的宏定义 (符号常量)	201
7.3 枚举	177	9.1.2 带参数的宏定义	203
7.3.1 枚举的定义	177	9.2 文件包含	204
7.3.2 枚举变量的说明	177	9.2.1 文件包含的一般形式	204
7.3.3 枚举类型变量的赋值和使用	178	9.2.2 文件包含的几点说明	205
7.4 类型定义符 typedef	178		



9.3 条件编译	206	10.2 使用位域	214
9.3.1 常用条件编译命令的形式	206	10.2.1 位域的定义	214
9.3.2 其他一些标准指令	208	10.2.1 位域变量的声明	214
9.4 本章小结	208	10.2.2 位域的使用	215
9.5 思考与练习	208	10.3 典型例题解析	216
一、填空题	208	10.4 本章小结	217
二、选择题	209	10.5 思考与练习	218
第 10 章 位运算	210	一、填空题	218
【本章概览】	210	二、选择题	218
【本章重点】	210	三、算法设计	218
【本章目标】	210	四、编程题	218
10.1 位运算符和位运算	210	附录 1 ASCII 字符与编码对照表	219
10.1.1 按位与运算	211	附录 2 运算符和结合性	220
10.1.2 按位或运算	211	附录 3 C 常用库函数	221
10.1.3 按位异或运算	212	附录 4 C 语言中的 32 个关键字	225
10.1.4 求反运算	212	附录 5 C 语言常用语法提要	227
10.1.5 左移运算	213	附录 6 C 语言库函数	231
10.1.6 右移运算	213		

第 1 章 C 语言基本知识

【本章概览】

C 语言是在上世纪 70 年代问世的，是目前最为流行的编程语言之一。随着 C 语言应用的不断深入，它又派生出多种版本，如 Microsoft C（简称 MC）、Borland Turbo C（简称 TC）。在上世纪 80 年代，随着面向对象程序设计思想的流行，又在 C 语言的基础上产生了 C++ 语言。同样 C++ 语言也有多种版本，如 Borland C++、Microsoft Visual C++ 等。

【本章重点】

- C 语言的基本知识
- C 语言的基本组成
- Turbo C 2.0 的上机步骤

【本章目标】

- 了解计算机编程语言和语言的特点
- 掌握 C 程序的基本组成
- 了解并配置 Turbo C 2.0 的集成环境，对源程序进行编译、链接并运行

1.1 C 语言概述

计算机是能够以比人快几百万甚至几十亿倍的速度执行计算和逻辑判断的设备，它在被称为“计算机程序”的指令集控制下处理数据。计算机程序控制着计算机，使它按顺序执行一系列动作，而设计和编制这些程序的过程就是程序设计。

1.1.1 计算机编程语言

计算机语言是程序设计的最重要的工具，它是指计算机能够接受和处理的、具有一定语法规则的语言。从计算机诞生至今，计算机语言经历了机器语言、汇编语言和高级语言几个阶段。C 语言就是一种面向过程的计算机高级编程语言。

在所有的程序设计语言中，只有机器语言编制的源程序能够被计算机直接理解和执行，



用其它程序设计语言编写的程序都必须利用语言处理程序“翻译”成计算机所能识别的机器语言程序。

针对不同的程序设计语言编写出的程序，语言处理程序一般有两种不同的处理形式：解释方式和编译方式。早期的一些高级语言采用解释程序进行翻译，如 BASIC，现在大部分高级语言都是采用编译程序进行翻译的，C 语言就是其中之一。C 语言开发程序的过程如图 1-1 所示。

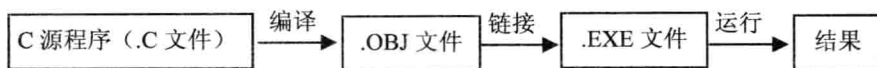


图 1-1 C 语言程序开发的过程

1.1.2 C 语言的发展历史

C 语言是 1972 年至 1973 年间由贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出的，其原型是 ALGOL60 程序设计语言。

1960 年出现的 ALGOL60 语言离硬件较远，不适宜编写系统程序。1963 年英国的剑桥大学在 ALGOL60 的基础上推出了 CPL 语言，它接近硬件一些，但规模较大，难以实现。1967 年剑桥大学的 Martin Richards 对 CPL 语言进行简化，推出了 BCPL 语言，但这种语言只具有单一的数据类型，处理能力可想而知。

1970 年美国贝尔实验室的 Ken Thompson 在 BCPL 的基础上设计出了简单而且接近硬件的 B 语言，并用 B 语言设计出了第一个 UNIX 操作系统。但 B 语言过于简单，功能有限，于是贝尔实验室的 D.M.Ritchie 在 B 语言的基础上引进多种数据类型，设计出了 C 语言。

1973 年，贝尔实验室的 K.Thompson 和 D.M.Ritchie 合作用 C 语言重写了 UNIX 操作系统，即 UNIX 第五版。从此 C 语言被用来代替汇编语言设计系统程序，大大促进了系统软件的开发，从而受到了程序员的普遍欢迎，迅速风靡世界。

1983 年，美国国家标准化协会（ANSI）根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了一个标准，称为 ANSI C。1987 年，ANSI 又公布了新标准——87 ANSI C。目前流行的 C 编译系统都是以它为基础的，本书的叙述也将以 87 ANSI C 为标准。

C 语言是一种面向过程的语言，在 20 世纪 80 年代提出了面向对象的程序设计（即 OOP），在这种形势下，贝尔实验室的 Bjarne Stroustrup 博士及其同事在 C 语言的基础上开发出了 C++ 语言。它是对 C 的扩充，是 C 的超集，它保留了 C 语言原有的所有优点，增加了类、对象等面向对象的机制。面向对象程序设计方法主要是解决大型软件的设计问题，因此，我们现在应掌握好 C 语言的程序设计，为将来进一步学习 C++ 打好基础。

1.1.3 C 语言的特点

计算机语言经历了机器语言、汇编语言和高级语言几个阶段。机器语言可以被计算机直接识别，但它是用二进制代码写成的，一般人难以学习掌握；汇编语言引入了一些英文助记



符,但针对不同型号的机器需要写出不同的代码,也就是说写出的程序移植性差。机器语言和汇编语言也被统称为低级语言。而C语言则既具有了接近人类自然语言的高级语言的特点,又具有低级语言的一些特点,如能进行位操作,可直接对硬件进行操作等。因此也有人把C语言称为“中级语言”。

C语言是一种理想的结构化程序设计语言,符合现代编程风格的要求,提供了顺序、选择(分支)和循环三种基本结构的控制语句。用C语言编写的源程序具有较好的通用性和可移植性,基本上不做修改就能用于各种型号的计算机和各种操作系统。C语言的语法限制少,书写形式灵活,设计自由度高,但这也对编程者有了较高要求,很多时候需要编程者自己找出程序中的错误。因此,我们在学习C语言时,一定注意不要过分依赖机器。

1.2 C语言程序的结构

用C语言语句编写的程序称为C程序或C源程序。一个完整的C语言程序是由一个main函数(又称主函数)和若干个其他函数结合而成的,或仅有一个main函数构成。C程序的结构可以通过下面几个简单的C源程序来了解。

【例 1.1】

```
#include "stdio.h"
main ()
{
    printf("Hello,World!");
}
```

这是一个最简单的C源程序,该程序经过编译、链接并运行后会在屏幕上输出:

Hello,World!

程序第一行是一个预处理命令,第二行的main()是主函数名。C语言的源程序都是由函数组成的,任何一个源程序都必须有且只能有一个main函数,也就是主函数。一对花括号标志函数的开始和结束,花括号中间的若干语句构成函数体。

在【例 1.1】中,函数体中只有一个语句,即printf函数调用语句,printf是由系统提供的库函数,其作用是输出数据。在这条语句最后有一个分号,要注意C语言中的每条语句都要以分号结束。

【例 1.2】

```
#include "stdio.h"
main ()
{int a,b,add; /*定义三个 int 类型的变量*/
    a=12; b=34;
    add=a+b; /*将 a 与 b 之和放在变量 add 中*/
}
```



```
printf("a+b=%d\n",add);  
}
```

本程序中在两个语句后面出现了`/*.....*/`，它表示注释，在编写程序时，我们可以在任何地方加入注释来帮助我们和其他人阅读程序，也就是提高程序的可读性。但是，注释仅用于说明，对程序的编译和运行是不起作用的。

在这个程序中仍然是只包含了一个函数，所以它的名字一定是 `main`。在函数体中第一个语句是定义变量，`a`、`b` 和 `add` 分别是三个变量的名字，接下来的三个语句分别给三个变量赋值，也叫赋值语句。

在【例 1.2】中，三个语句写在了两行里，也就是说，C 语言的书写格式是很灵活的，一行中允许写多个语句，也可以将一个语句分成多行写，只要记住每条语句最后以分号结束，因为 C 语言是以分号而不是回车或其他分隔符来判断语句结束的。

最后一个语句仍然是 `printf` 函数调用语句，但可以和【例 1.1】比较一下不同。在 C 语言中，函数名后面都会跟一对小括号，象上面的 `main()` 和 `printf()`。小括号里是函数的参数，如果小括号里是空的，就表示这是一个无参函数，如 `main()`；而有些函数一定要有参数，如 `printf()`。

在【例 1.1】中，`printf` 函数的参数是用一对双引号引起来的，被称为字符串，它在 `printf` 函数中会被原样输出，像我们上面看到的那样。而在【例 1.2】中，字符串中的 `%d` 和 `\n` 就不再原样输出了，`%d` 被称为格式字符，它表示要把和它对应的数据以十进制整数形式输出，此例中为变量 `add`，而 `\n` 被称为“转义字符”，它表示回车换行。因此，【例 1.2】的源程序经过编译、链接并运行后在屏幕上的输出结果为：

```
a+b=46
```

并且光标自动换到下一行的开头位置。

【例 1.3】

```
#include "stdio.h"  
  
int sum (int x, int y)  
{int z;  
  z=x+y;  
  return z;  
}  
  
main ()  
{int a, b, add;  
  a=12; b=34;  
  add=sum (a, b);  
  printf("a+b=%d\n",add);  
}
```

这个源程序的作用和输出结果都和【例 1.2】完全一样，但它是用两个函数来完成的，一个是 `main` 函数，还有一个是我们自己定义的函数，名字叫 `sum`。`main` 函数在这里又被称



为主调函数，而 `sum` 函数则称为被调函数。`sum` 函数有两个参数 `x` 和 `y`，它们的值由主调函数传递过来。请注意 `main` 函数中的 `add=sum(a, b)`；语句，它的作用是把 `a` 和 `b` 的值通过调用 `sum` 函数传递给 `x` 和 `y`，求出的和再通过 `sum` 函数中的 `return z`；语句返回到这里并赋值给 `add`，从而得到和【例 1.2】完全一样的输出结果。

通过以上三个例子可以看到：

(1) 一个 C 源程序至少包含一个 `main` 函数，或者包含一个 `main` 函数和若干个其他函数。也就是说，C 源程序是由函数构成的，函数是 C 源程序的基本单位。其他函数可以是系统提供的标准库函数，也可以是用户根据实际需要自己设计编写的函数。

(2) 一个函数的基本结构为：

函数类型 函数名 (函数参数类型 函数参数名, ...)

{ 声明部分;

执行部分;

}

其中声明部分和执行部分合在一起又被称为函数体。

在 C 语言中，也允许函数体是空的，这种函数被称为空函数。如：`dump()`

{ }

空函数什么也不做，但它是合法的。我们在设计调试一些大型程序时可以在某些地方放置一个空函数。

(3) C 程序的执行都是从 `main` 函数开始，并且一定结束于 `main` 函数，而不管 `main` 函数在程序中的位置如何。

(4) 函数体中的每一个语句都要以分号结束。C 语言的书写格式是非常自由的，我们可以把多个语句写在一行上，也可以把一个语句分写有多行上，系统是以分号判断一个语句结束的。

(5) 在程序的任何地方都可以加入以 “/*” 和 “*/” 包围起来的注释，注释的作用是为了增加程序的可读性，它并不被系统执行。

(6) C 语言中大小写字母是严格区分的。例如：`main` 如果任何一个字母写成大写就是错的。这一点我们在使用时要特别注意。

1.3 运行 C 程序的步骤与方法

在微型机上，常用的 C 语言编译系统有 Microsoft C、Turbo C、Borland C 等，它们之间以及它们的不同版本之间都略有差异，但它们的基本部分都是符合 ANSI C 标准的。今后在使用时要了解所用的计算机系统所配置的 C 编译系统的一些规定。下面首先介绍一下使用 Turbo C 2.0 的上机步骤。

1.3.1 进入 Turbo C 2.0 集成环境

首先找到 Turbo C 2.0 所在的文件夹（子目录），在该文件夹下运行 Turbo C 2.0 的执行程



序 TC.EXE，屏幕上就会出现如图 1-2 的 TC 主屏，表示进入了 Turbo C 2.0 集成开发环境。

从图中可以看出，TC 主屏由四部分组成：主菜单、编辑窗口、信息窗口和功能键提示行。主菜单上的每个命令又都包含一个下拉子菜单（Edit 除外），编辑窗口是用来输入或装载源程序的，信息窗口用于查看编译和调试程序时的一些信息。

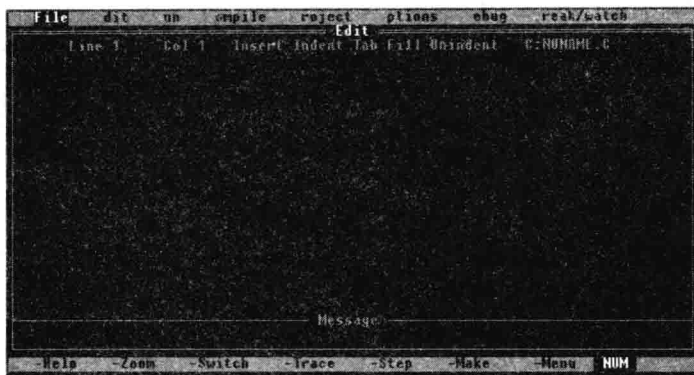


图 1-2 Turbo C 2.0 的主屏

1.3.2 配置运行环境

配置运行环境，也就是设置好包含文件、库文件和 Turbo C 系统所在的目录，是非常关键的一步。如图 1-3 所示。

- 在配置之前要先知道当前所用的 Turbo C 2.0 系统所在的路径，假定当前所用的 Turbo C 2.0 被装在 C 盘的 APPL 目录下，目录名为 TC（即路径为 C:\APPL\TC）。
- 选择 Options>Directories 命令，打开图 1.3 所示对话框，依次在“Include directories:”、“Library directories:”和“Turbo C directories:”后输入“C:\APPL\TC\INCLUDE”、“C:\APPL\TC\LIB”和“C:\APPL\TC”。
- 按【Esc】键返回后，选择 Options>Save Options 命令，保存刚才的设置到“tcconfig.tc”文件中。

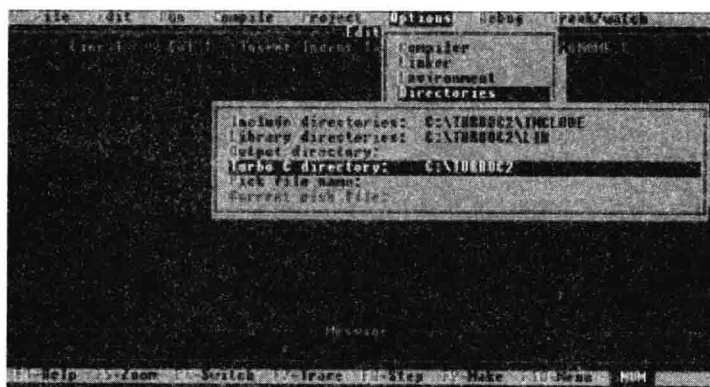


图 1-3 配置 Turbo C 2.0 运行环境



1.3.3 编辑源程序

编辑源程序，就是在编辑窗口中将事先编写好并已经自查过的源程序输入计算机或对已经存在的源程序文件进行修改的过程。如果是要输入一个新程序，应首先选择 File>New 菜单创建新文件。输入程序后可选择 File>Save 菜单进行保存。

如果是要载入一个已经存在的源程序文件，可选择 File>Load 菜单，在出现的对话框中选择要载入的源程序文件名。对文件进行修改后，一定要记住保存。除了可以选择 File>Save 菜单以原来的名字保存外，还可以选择 File>Write to 菜单进行换名保存。

1.3.4 对源程序进行编译、链接并运行

一个 C 源程序是不能直接运行的，它必须要由系统进行编译、链接后才能运行。在 Turbo C 2.0 中，可以将编译、链接、运行一步一步来做，也可以将这三步合并完成。此时，只要选择 Run>Run 命令或直接按快捷键【Ctrl+F9】就可以了。

在编译程序时，如果源程序有问题的话，系统会在信息窗口显示相应的提示信息，这时可以回到上一步对源程序重新编辑保存。这里需要注意以下两点：

(1) 程序的运行结果并不在这个窗口显示，不要因看不到运行结果就反复按【Ctrl+F9】键，否则会看到一堆重复的运行结果。

(2) Turbo C 2.0 的错误提示信息是一个比较模糊的信息，需要认真阅读源程序，不要因为看到信息窗口里出现了一堆错误信息就失去信心，有时候一大堆错误信息后面其实只是一处小错误。

1.3.5 查看运行结果

通过前面的介绍我们已经知道，在 Turbo C 2.0 中，程序的运行结果和源程序并不在一个窗口中显示。在我们按下【Ctrl+F9】键时，如果程序不需要输入数据并且程序语法完全正确的话，集成环境的主屏会闪烁一下，这时程序的运行结果已经有了，那怎么看到呢？只需要按【Alt+F5】键或选择 Run>User Screen 命令，便切换到了用户屏。

如果程序需要输入数据，则在我们按下【Ctrl+F9】键后，屏幕就处于用户屏状态，等待用户正确输入所需的数据，输入结束程序运行后仍会回到集成环境的主屏，还要按上面介绍的方法切换到用户屏查看程序的运行结果。

这里也要提醒大家：不要以为有了运行结果程序就认为程序一定正确。因为系统只会给我们检查出一些语法错误，而程序中的一些逻辑性错误是不检查不出来的，所以我们要认真分析一下运行结果是否符合我们的要求。

1.3.6 退出 Turbo C 2.0 集成环境

一个程序运行完成后，我们可以选择 File>New 菜单或者 File>Load 菜单开始下一个程序的调试，千万不要把两个程序输入在同一个编辑窗口。



所有的程序调试完成后,选择 File>Quit 命令彻底退出 Turbo C 2.0,之后若再想进入 Turbo C 2.0 集成环境,则需要重新运行可执行文件 TC.EXE。

此外,也可选择 File>OS shell 菜单暂时退出 Turbo C 2.0,转到 DOS 提示符下。此后若想再回到 Turbo C 2.0 集成环境的主屏,只需在 DOS 提示符下键入 EXIT 即可。

以上只是简要了 Turbo C 2.0 集成环境的基本使用方法,在 Turbo C 2.0 集成环境的主菜单下还有许多命令可以帮助我们对源程序进行调试,大家可以参考有关书籍试一试。

当然,只要我们勤动脑动手,上机时细心加耐心,一定会学好 C 语言,成为一个编程高手的。

1.4 本章小结

本章主要讲述了 C 语言的基本知识、C 语言程序的结构,以及运行 C 程序的步骤与方法。通过本章的学习,读者应了解计算机编程语言和语言的特点,掌握 C 程序的基本组成,了解并清楚配置 C 语言的集成环境,对源程序进行编译、链接并运行等方面的知识。

1.5 思考与练习

一、填空题

1. C 程序都是由_____构成的,其中必有一个且只有一个_____。
2. C 程序是从_____开始执行的,最后也结束于_____。

二、操作题

1. 试写出函数的一般形式。
2. 已知一个学生的三门课的成绩分别为: 76, 95, 89, 试编写一个程序求该学生的总成绩和平均成绩,并上机验证程序的正确性。