

高等职业技术教育计算机系列教材

C 语言程序设计



实用教程

杨振生 张玉荣 编著

C YUYAN CHENGXU SHEJI
SHIYONG JIAOCHENG

C YUYAN CHENGXU SHEJI
SHIYONG JIAOCHENG



中国科学技术大学出版社

高等职业技术教育计算机系列教材

C 语言程序设计

实用教程

杨振生 张玉荣 编著

江苏工业学院图书馆
藏书章

中国科学技术大学出版社

2004·合肥

内 容 提 要

本书系统地介绍了C语言的基本内容及其程序设计方法,深入浅出,通俗易懂。根据高等职业技术教育的特点和培养目标的要求,本书强调基本技能的培养,突出应用性和实效性,增强可读性。

全书共分十章,主要包括:C语言概述、数据类型与运算符、顺序结构、选择结构、循环结构、数组、指针、函数、结构体与共用体、文件等。书中例题丰富,侧重基本概念和程序设计方法。每章最后都附有习题和实习参考题。在教学实施过程中,供教师选择或参考。

本书可作为高职高专院校的计算机、信息工程、电子类专业的教材,也可作为自学读物。

图书在版编目(CIP)数据

C语言程序设计实用教程/杨振生,张玉荣编著. —合肥:中国科学技术大学出版社, 2004. 10

ISBN 7-312-01722-3

I. C… II. ①杨… ②张… 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(2004)第 079187 号

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路96号,邮编:230026,发行电话:0551-3602905,3602906)

合肥学苑印务有限公司印刷

全国新华书店经销

开本:787×1092mm 1/16 印张:15.125 字数:380千

2004年10月第1版 2004年10月第1次印刷

印数:1—4000册

ISBN 7-312-01722-3/TP·347 定价:19.00元

前 言

目前,我国高等教育事业正在蓬勃发展,特别是高等职业教育异军突起,其发展速度之迅猛,势头之强劲,远远超出了人们的预料,给我国高等教育事业注入了新的生机和活力。根据统计高等职业教育的学生规模已占据了全国整个高等教育的半壁江山。这说明高等职业教育和普通高等教育在规模上可相提并论,在我国高等教育体系中占有相当重要的地位。

高等职业教育是一个新生事物,它开创了一种新的办学模式,其培养目标是具备必要的理论基础和较强的实践能力,直接面向工作在生产、管理和服务于第一线的应用型、技能型的高级实用人才。根据这一培养目标,传统的高等院校教材已不适应高职教育的需要。高职教材必须符合培养目标的要求,逐渐形成自身的特色。因此,当前高职教材建设是一项刻不容缓的迫切任务。

目前各类高职教材陆续出现,实际上都处于探索阶段。特别是在以“应用型”、“技能型”为主旨和特征的教材体系建设上,要走一条前人没有走过的路。在既符合培养目标,又符合社会实际需要的结合点上要做许多艰苦的工作。本书的编写,实际上也是在这方面进行一种初步探索。

全书共分10章。第1章是C语言概述;第2章介绍了C语言的数据类型与运算符;第3、4、5章分别介绍了顺序结构、选择结构和循环结构;第6、7、8、9章分别介绍了数组、指针、函数、结构体与共用体;第10章介绍了文件。其中第1章至第5章由张玉荣编写,第6章至第10章由杨振生编写。

由于作者水平有限,书中不妥和疏漏之处在所难免,敬请专家和读者指正。

编 者

2004年7月

目 录

第 1 章 C 语言概述	1
1.1 计算机语言和计算机程序	1
1.1.1 计算机语言	1
1.1.2 计算机程序	2
1.2 C 语言的发展及其特点	3
1.2.1 C 语言的发展	3
1.2.2 C 语言的特点	3
1.3 简单的 C 程序介绍	4
1.4 C 语言程序的运行环境	6
1.4.1 Turbo C(TC) 集成开发环境	6
1.4.2 C 语言程序的上机步骤	9
习题	12
实习参考题	13
第 2 章 数据类型、运算符与表达式	14
2.1 C 语言的数据类型	14
2.2 常量	15
2.2.1 整型常量	15
2.2.2 实型常量	17
2.2.3 字符型常量	17
2.2.4 字符串常量	18
2.2.5 符号型常量	18
2.3 变量	19
2.3.1 变量的数据类型及其定义	20
2.3.2 变量的存储类型	21
2.3.3 变量的初始化	21
2.4 运算符与表达式	22
2.4.1 C 运算符与表达式简介	22
2.4.2 算术运算符及其表达式	23
2.4.3 关系运算符及其表达式	24
2.4.4 逻辑运算符及其表达式	25
2.4.5 赋值运算符及其表达式	26
2.4.6 逗号运算符及其表达式	27

2.4.7 条件运算符及其表达式	28
2.4.8 位运算符和位运算	29
2.4.9 表达式中数据类型的转换	30
习题	32
实习参考题	34
第3章 顺序结构程序设计	35
3.1 结构化程序设计的基本概念	35
3.2 C 语句概述	38
3.3 单字符数据的输入/输出	39
3.3.1 单字符数据的输出函数 putchar()	39
3.3.2 单字符数据的输入函数 getchar()	40
3.4 格式输入/输出	40
3.4.1 格式输出函数 printf()	41
3.4.2 格式输入函数 scanf()	42
3.5 顺序结构的程序设计举例	44
习题	45
实习参考题	48
第4章 选择结构程序设计	49
4.1 if 语句	49
4.2 if 语句的嵌套	51
4.2.1 if—if—else 语句	51
4.2.2 if—else—if 语句	52
4.3 switch 语句	52
4.4 选择结构的程序设计举例	53
习题	56
实习参考题	60
第5章 循环结构程序设计	62
5.1 goto 语句以及用 goto 语句构成的循环	62
5.2 while 语句和 do~while 语句	63
5.2.1 while 语句	63
5.2.2 do~while 语句	64
5.2.3 两种循环的比较	65
5.3 for 语句	66
5.4 循环的嵌套	68
5.5 break 语句和 continue 语句	70
5.5.1 break 语句	70

5.5.2 continue 语句	70
5.6 循环结构的程序设计举例	72
习题	73
实习参考题	77
第 6 章 数组	79
6.1 数组的概念	79
6.2 一维数组	80
6.2.1 一维数组的定义	80
6.2.2 一维数组的初始化	80
6.2.3 一维数组元素的引用	81
6.2.4 一维数组应用举例	81
6.3 二维数组	83
6.3.1 二维数组的定义	83
6.3.2 二维数组的初始化	84
6.3.3 二维数组元素的引用	84
6.3.4 二维数组应用举例	85
6.4 字符数组与字符串处理	87
6.4.1 字符数组的定义	87
6.4.2 字符数组的初始化	87
6.4.3 字符数组的引用	88
6.4.4 字符串处理函数	89
6.4.5 字符数组应用举例	93
习题	95
实习参考题	99
第 7 章 指针	101
7.1 指针的基本概念	101
7.2 指针变量	102
7.2.1 指针变量的定义	103
7.2.2 指针变量的初始化	103
7.2.3 指针变量的引用	103
7.3 指针变量的运算	105
7.4 指针与数组	108
7.4.1 指向一维数组的指针变量	108
7.4.2 通过指针变量引用数组元素	108
7.4.3 指向二维数组的指针变量	111
7.5 字符指针与字符串	115
7.5.1 指向字符串常量的指针变量	115

7.5.2	字符指针的引用	116
7.5.3	指向字符数组的指针变量	116
7.6	指针数组	117
7.6.1	指针数组的定义与初始化	117
7.6.2	指针数组元素的引用	118
7.7	多级指针	120
7.8	带参数的主函数	122
7.9	指针应用程序设计举例	123
	习题	126
	实习参考题	130
第 8 章 函数		
8.1	模块化程序设计的基本概念	131
8.2	有关函数的概念	132
8.3	函数的定义与调用	133
8.3.1	函数的定义	133
8.3.2	函数的调用	135
8.4	函数间的数据传递	139
8.4.1	值传递方式	139
8.4.2	地址传递方式	141
8.4.3	返回值传递方式	143
8.4.4	全局外部变量传递方式	143
8.5	数组作为函数的参数	144
8.5.1	数组元素作为函数的参数	145
8.5.2	数组名作为函数的参数	145
8.6	函数的嵌套调用与递归调用	146
8.6.1	函数的嵌套调用	147
8.6.2	函数的递归调用	148
8.7	指针型函数和指向函数的指针	150
8.7.1	指针型函数	151
8.7.2	指向函数的指针	152
8.8	变量的存储方式及其作用域与初始化	153
8.8.1	变量的存储方式	153
8.8.2	变量的生存期与作用域	154
8.8.3	变量的初始化	157
8.9	编译预处理	158
8.9.1	宏定义	159
8.9.2	文件包含	162
8.9.3	条件编译	163

习题	165
实习参考题	172
第 9 章 结构体、共用体与枚举型	173
9.1 结构体类型与结构体变量	173
9.1.1 结构体类型的定义	173
9.1.2 结构体变量的定义	174
9.1.3 结构体变量的引用	176
9.1.4 结构体变量的初始化	176
9.2 结构体数组	177
9.2.1 结构体数组的定义	177
9.2.2 结构体数组的初始化	179
9.3 结构体指针	180
9.3.1 结构体指针的定义与引用	180
9.3.2 指向结构体数组的指针	181
9.3.3 利用结构体指针在函数间传递结构体	182
9.4 结构型函数和结构指针型函数	184
9.4.1 结构型函数	184
9.4.2 结构指针型函数	185
9.5 结构体嵌套	186
9.6 共用体	188
9.6.1 共用体类型的定义	188
9.6.2 共用体变量的定义	189
9.6.3 共用体变量的引用	190
9.7 枚举类型	192
9.7.1 枚举型的定义	192
9.7.2 枚举型变量的定义	193
9.7.3 枚举型变量的引用	193
9.8 用户自定义类型	194
习题	196
实习参考题	201
第 10 章 文件	202
10.1 文件的概述	202
10.1.1 文件的概念	202
10.1.2 缓冲文件系统	203
10.1.3 文件的分类	203
10.1.4 文件型指针	204
10.1.5 文件的处理过程	204

10.2	文件的打开与关闭	205
10.2.1	文件的打开	205
10.2.2	文件的关闭	206
10.2.3	标准设备文件	206
10.3	文件的读写	207
10.3.1	文件尾测试函数	207
10.3.2	字符读/写函数	207
10.3.3	字符串读/写函数	209
10.3.4	数据块读/写函数	210
10.3.5	格式化读/写函数	211
10.4	文件的定位与错误检测	212
10.4.1	文件头定位函数	213
10.4.2	文件随机定位函数	213
10.4.3	文件当前读/写位置函数	214
10.4.4	错误检测函数	215
	习题	215
	实习参考题	218
附录一	ASCII 码表	219
附录二	运算符及其优先级和结合性	220
附录三	Turbo C 2.0 常用库函数	221
附录四	常见错误信息表	228
	参考文献	232

第1章 C语言概述

本章重点:(1)计算机语言和计算机程序;(2)C语言的产生及其特点;(3)C语言程序的运行环境。

本章要求:(1)理解计算机语言和计算机程序的概念;(2)了解C语言产生的历史背景及其特点;(3)初步熟悉C程序的特性;(4)掌握C语言上机方法与步骤。

当今世界随着计算机技术的迅猛发展,人和计算机之间交往的“语言”也在不断更新。从六十年代初期出现的 ALGOL 60、COBOL 等语言,发展到七十年代 C 语言的诞生。C 语言自问世以来,表现出了很强的生命力。因为 C 语言简洁、表达能力强、使用方便灵活、产生目标代码质量高、可移植性好,它既可以用来编制系统软件,也可以编制应用软件,所以到目前为止,C 语言仍是国际上深受欢迎的主流计算机语言之一。

本章将从计算机语言和计算机程序讲起,具体介绍 C 语言的发展及其特点,给出 C 程序的框架,并详细阐述运行一个 C 程序的完整过程。

1.1 计算机语言和计算机程序

计算机为什么能处理各种不同的问题呢?这是由于人们事先对各类问题进行了分析,确定了解决问题的方法和步骤,然后根据它使用某一种计算机语言(如 C 语言)编写出相应的计算机程序,再让计算机执行这个程序,才能得到最后的结果。因此计算机语言是编制计算机程序的一个工具。

1.1.1 计算机语言

从世界上第一台计算机诞生以后,多年来,随着计算机技术的不断更新,计算机语言也经历了几代的发展。

1. 机器语言

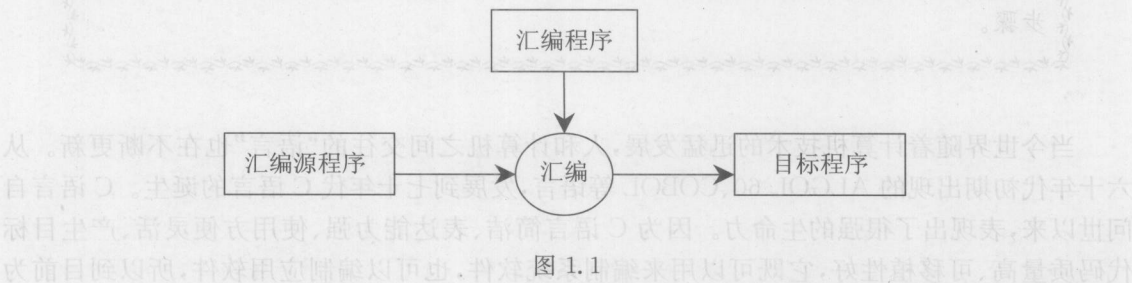
人和计算机交往的一个主要障碍则是“语言”。计算机根本无法理解人类的自然语言(如英文、中文),而只能接受并执行由 0、1 构成的二进制的指令,计算机所有的操作都是在指令的控制下进行的。计算机能够直接识别和执行的这种指令,称为机器指令,每一种类型的计算机都规定了自己的一套指令,这种指令的集合就是机器语言指令系统(简称机器语言)。例如,有的计算机的指令系统包含 100 多种指令,为解决某一特定问题,需要选用指令系统中某些指令,这些指令序列就组成一个“程序”,即所谓的机器语言程序。可是机器语言程序难理解、难

修改给用户带来很大不便。

2. 符号语言(汇编语言)

符号语言即汇编语言是从机器语言发展演变而来的,它用一些“助记符号”来代替那些难理解的二进制代码。例如,用 SUB A,B 代表将 A 累加器中的数和 B 寄存器中的数相减,其结果放在累加器 A 中。用 ADD A,B 代表一次加法,显然,这种表示形式要比机器语言更容易理解和使用。

计算机并不能直接执行符号语言程序,必须先把他们逐条翻译成机器指令,然后交计算机执行,这种翻译操作称为“汇编”,是由一种专门的“汇编程序”来实现这种转换工作的。因此符号语言又称为“汇编语言”。用汇编语言(或高级语言)编写的程序称为源程序(source program),经过转换后具备了机器语言的形式称为目标程序(object program)。其转换关系如图 1.1 所示。



符号语言和机器语言的指令基本上是一一对应的,它们都是针对特定的计算机系统的,不同类型的计算机所用的符号语言也不同。所以我们称机器语言和汇编语言为“面向机器的语言”,它们也被称为“低级语言”。

3. 高级语言

高级语言与具体计算机内部结构无关,即用它所写的程序可以在任一台上计算机上运行(必要时作一些小的改动)。这种语言称为“面向过程的语言”,只需要根据问题求解的步骤,写出处理过程即可。

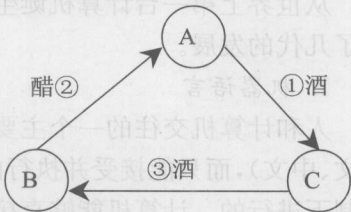
“面向过程的语言”属于“高级语言”。例如,从 60 年代相继出现的 COBOL、ALGOL60、BASIC 等语言。学习使用高级语言要比低级语言容易得多。例如,BASIC 语言中用“IF X>0 THEN PRINT X ELSE PRINT -X”来表示“如果 X>0 打印 X 的值,否则打印 -X 的值”。显然,这种表示方法很容易理解。因此,学习使用高级语言为计算机的推广普及扫除了一大障碍。即使对计算机内部结构毫无所知的人,也能学会使用高级语言来编写程序去解决面临的问题。

1.1.2 计算机程序

程序是解决某一个具体问题的方法步骤,而计算机程序则是用某种计算机语言来描述解决一个具体问题的方法、步骤。

【问题 1】 有两个杯子 A 和 B 分别盛放酒和醋,要求将它们互换,互换后 A 杯子里盛放醋,B 杯子里盛放酒。

根据日常经验,必须增加一个空杯子 C 作为过渡,见图 1.2 所示。



第一步: 先将 A 杯子中酒倒入 C 杯子

第二步: 再将 B 杯子中醋倒入 A 杯子

第三步: 最后将 C 杯子中酒倒入 B 杯子

上述步骤就是一个解决交换问题的程序。

【问题 2】 设某班有 50 名学生的 C 语言课程考试成绩, 求其平均成绩。

经分析如果班级有 50 人, 则从键盘输入 50 个数作为学生成绩, 以求出平均成绩。根据题意, 我们先设 S 代表成绩总和, T 代表每一次输入的数值, 用 I 代表输入数的次数。则步骤如下:

第一步: 使 $S=0$, 输入第 1 个数值给 T, $I=1$

第二步: 使 $S+T \Rightarrow S$, 得到的和仍放在 S 中

第三步: 再输入下一个数给 T, 使 I 增 1

第四步: 判 I 是否小于或等于 50, 如果小于或等于 50, 则返回执行第二步; 否则停止循环执行, 此时 S 中放的是 50 个学生的成绩总和

第五步: 用 S 除以 50, 即得到 50 个学生的平均成绩

上述步骤就是一个解决对若干个数求平均数问题的程序。

以上两个程序仅用文字描述出来, 那么如何用计算机语言——C 语言来描述这两个问题呢? 我们将在后面给出。

1.2 C 语言的发展及其特点

C 语言是目前世界上广为流行的计算机语言之一。它既适合编制系统程序, 也适合编制应用程序。

1.2.1 C 语言的发展

C 语言由 Dennis Ritchie 设计, 并首次在一台使用 UNIX 操作系统 DEC PDP-11 计算机上实现。C 语言是由一种早期的编程语言 CPL 发展演变而来。1967 年英国剑桥大学的 Martin Richards 对 CPL 语言作了简化, 推出了 BCPL 语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础, 作进一步简化, 设计出 B 语言。随着 B 语言的发展, 最后导致了 70 年代 C 语言的问世。

随着微型计算机的日益普及, 大量的 C 语言版本相继出现。尽管这些版本之间几乎可以高度兼容, 但由于没有统一的标准。为了改变这种情况, 到 1983 年, 美国国家标准化协会 (ANSI) 根据 C 语言问世以来的各种版本为 C 语言制定了 ANSI 标准, 称为 ANSI C, 该标准于 1987 年又进行了改善。目前流行的各种 C 编译系统如 Microsoft C、Quick C、Turbo C 都是以 87 ANSI C 为基础的。

1.2.2 C 语言的特点

C 语言能够成为目前世界上广泛流行的语言之一, 必定有其不同于其他语言的特点。C 语言的主要特点如下:

(1) C 语言是中级计算机语言。它把高级语言 (如 BASIC) 的面向过程与低级语言接近硬

件的特性有机地结合起来,可以说它兼有高级语言和低级语言的优点。

(2) C 语言简洁、紧凑、使用方便灵活。C 语言共有 32 个标准关键字,9 种控制语句,压缩了一切不必要的成份。

(3) C 语言的运算符非常丰富。C 语言共有 34 种运算符,C 语言中把逗号、括号、等号、问号等都作为运算符的一部分来处理,通过运算符可以构成各式各样的表达式,能完成其他高级语言难以实现的运算。

(4) C 语言数据类型丰富。C 语言的常用数据类型有整型、实型、字符型等,并在此基础上构造出的各种复杂类型,如数组、结构体和共用体等。C 语言也不象有些语言那样有很强的数据类型的区分,它允许几乎所有的数据类型转换,例如,在所有表达式中,字符型和整型都可以混合使用。

(5) C 语言是结构化、模块化的编程语言。C 语言有三种基本结构:顺序结构、选择结构和循环结构,以实现程序设计的结构化。C 语言程序是通过函数的手段实现模块化。

(6) C 语言允许直接对位、字节、字和指针进行操作。所以它也适用于对系统程序的开发与研制。

(7) C 语言代码具有很好的可移植性。用 C 语言编写的程序,基本上不作修改就能用于各种型号的计算机和各种操作系统。

(8) C 语言程序生成的目标代码质量高,运行速度快,程序执行效率高。

总之,由于 C 语言具有以上特点,因此,受到广大用户的普遍欢迎。但由于 C 语言的语法限制不太严格,例如,C 语言程序在运行时不做有关数组边界检查和变量类型兼容性检查等。又由于 C 语言具有相当大的灵活性,因此也降低了程序的某些安全性,可能给初学者增加了一定难度。

1.3 简单的 C 程序介绍

下面将通过几个简单的 C 程序实例来具体介绍 C 程序的基本构成,以便读者对 C 程序的框架有一个初步的了解。在此基础上,再进一步分析 C 程序的特点。

例 1.1 源程序如下:

```
main( )
```

```
{
```

```
printf("Are you ready? \n");
```

```
printf("Ready! \n");
```

```
}
```

上述程序的执行结果为:

```
Are you ready?
```

```
Ready!
```

这个源程序只有一个称为主函数的 `main( )` 组成,函数体用一对大括弧 `{ }` 括起来,其间是两条输出语句,输出结果为双引号里的内容。其中“`\n`”是转义符(详见第 2 章),表示“换行”操作。

针对上一节 1.1.2 中给出的【问题 1】,用 C 语言如何实现交换问题。

例 1.2 源程序如下:

```
main()
{
    int a,b,c;          /* 定义 a,b,c 为整型变量 */
    a=10;b=20;          /* 给 a 和 b 赋初值 */
    c=a;a=b;b=c;        /* 通过 c 变量,实现交换 a,b 的值 */
    printf("a=%d,b=%d",a,b);
}
```

上述程序的执行结果为:

a=20,b=10

上述程序的作用是实现对 a 和 b 两个整型数的交换。这个源程序在主函数的函数体内包括两部分的内容:一部分是数据的定义,即数据的说明部分;另一部分则是 C 程序的执行语句,即语句执行部分。在源程序中,为了增加程序的可读性,可以用 /* */ 对源程序的任何部分作注释,通常注释部分不参与程序的编译和运行。

例 1.3 源程序如下:

```
float absx( float x)      /* 用户自定义函数 absx() */
{
    float y;              /* 定义 y 为实型变量 */
    if(x>0) y=x;
    else y=-x;             /* 如果 x>0,则将 x 值赋给 y,否则将 -x 值赋给 y */
    return(y);             /* 将 y 值返回给调用它的主函数 main() */
}

main()                    /* 主函数 */
{
    float x1,y1;           /* 定义 x1,y1 为实型变量 */
    scanf("%f",&x1);       /* 输入值给 x1 */
    y1=absx(x1);           /* 调用 absx() 函数,并将返回值赋给 y1 */
    printf("y1=%6.2f\n",y1); /* 输出 y1 */
}
```

上述程序执行情况为:

① 3.1 ✓

y1=□□3.10

② -4.5 ✓

y1=□□4.50

注:“□”表示空格,上述程序包括两个函数:主函数 main() 和自定义函数 absx(), 它们分别是主调用函数和被调用函数。其中 absx() 函数的作用是判断如果 $x > 0$, 则将 x 值赋给 y, 如果 $x \leq 0$, 则将 -x 的值赋给 y, 通过 return 语句将 y 的值返回给主调函数 main()。main() 函数中第 4 行为调用 absx() 函数,在调用时将主调函数中的参数 x1 值传递给被调用函数中的 x, 经过执行 absx() 函数后得到一个返回值,再把这个值赋给变量 y1, 然后按格式输出来。其中“%6.2f”表示输出是一个实型数值,且要求这个实型数值占 6 列,小数点后只能有 2 位(详见第 3 章中关于修饰符的说明部分)。

通过上述几个实例,可以看到 C 程序有如下几个特点:

(1) C 程序是由一个或多个函数组成的集合。但必须有且仅有一个主函数 main(), 程序

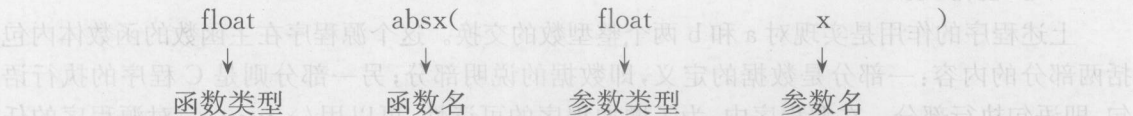
的执行从主函数开始,并在主函数中结束。程序中所使用的其他函数,既可以是系统提供的库函数(如 printf()、scanf() 函数),也可以是用户根据需要自己命名的函数(如例 1.3 中的 absx() 函数)。在 C 语言程序中,函数是基本的结构模块。C 的函数库十分丰富(见附录三),每一个函数在程序中都能完成一个独立的任务。因此,这个特点使得 C 语言很容易实现程序的模块化。

(2) 一个函数有两部分组成,即函数首部和函数体。

① 函数首部。它的一般结构如下:

[函数类型] 函数名(参数类型 参数名 1,参数类型 参数名 2,.....)

例如例 1.3 中的 absx() 函数的首部为:



注:()内可以为空,如:main();但函数名和一对圆括弧必须要有。

② 函数体即 C 语言的数据说明部分和语句执行部分。它的一般结构如下:

```
{  
    [数据说明部分]  
    [语句执行部分]  
}
```

注:在某些情况下也可以没有数据说明部分(如例 1.1),甚至{ }里可以什么都没有,如:

```
space()  
{  
}
```

在后续章节中将说明此函数是合法的,它是一个空函数,什么都不做。

(3) C 程序中每一个语句的最后总有一个分号,即分号是 C 语句的必要组成部分。

(4) C 程序中至少有一个或多个输出,可以有零个或多个输入。

(5) C 程序的书写格式自由,一行可以写多个语句,一个语句也可以写在多行上。

(6) 可以用 /* */ 对 C 程序中任何部分作注释,以增加程序的可读性,其中注释部分对程序的编译和执行没有影响。

1.4 C 语言程序的运行环境

在编写好一个完整的 C 源程序后,如何运行它并获得运行结果呢?这就需要我们建立一种 C 语言程序的运行环境,下面就 Turbo C 环境下如何运行一个 C 的源程序作具体介绍。

1.4.1 Turbo C(TC)集成开发环境

Turbo C 是美国 Borland 公司推出的,它是集源程序编辑、编译、调试、运行于一体的集成环境。它具有使用简单、方便、运行速度快、效率高、功能强等特点,C 语言编程者可以在 Turbo C 环境下完成几乎所有的工作。

在 Windows 操作系统下启动 Turbo C 软件后,屏幕上出现的是一个对话框,如图 1.3 所示。利用 Enter 键+空格键可以将 TC 全屏显示,屏幕上分别是主菜单、状态行、编辑窗口和

功能键提示行,如图 1.4 所示。

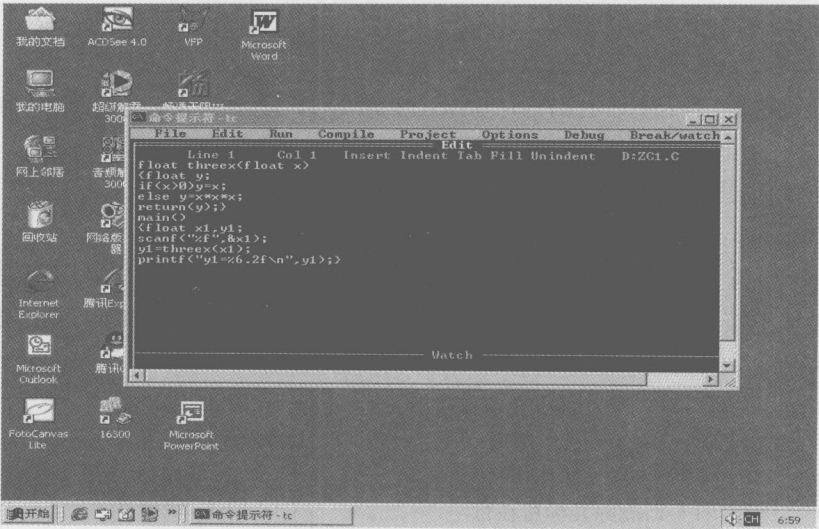


图 1.3 TC 对话框

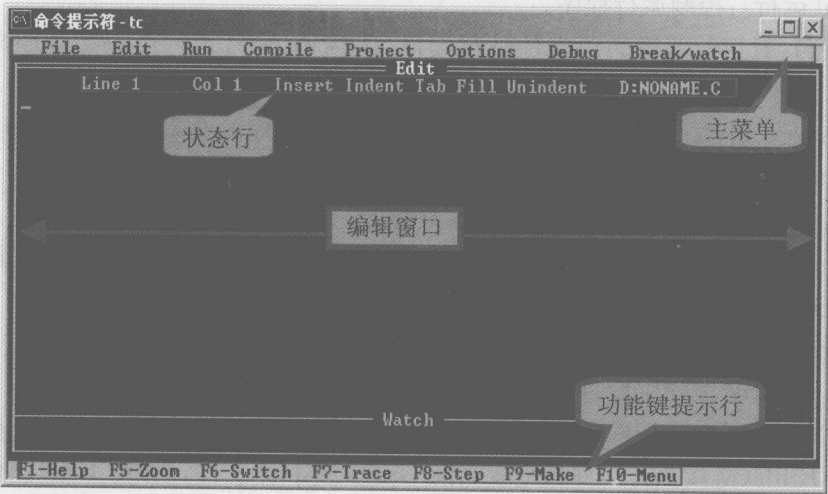


图 1.4 TC 窗口的介绍

1. Turbo C 主菜单的介绍

Turbo C 的主菜单共有 8 个菜单项：

(1) File(文件)：主要用于对文件的操作。

其下拉菜单有如下菜单项：

Load (加载)：装入一个文件,可以用通配符(*,?)来进行列表选择,也可以装入某一个文件,只要给出文件名即可。

Pick(选择)：最近装入编辑窗口的 8 个文件列成一个表,让用户选择。

New (新文件)：建立一个新文件,缺省文件名为 NONAME.C。

Save(存盘)：将编辑窗口里的文件存盘。