

高等学校电子信息类系列教材

C语言程序设计

顾治华 陈天煌 王 涛 编著

四川大学出版社



C 语言程序设计

顾治华 陈天煌 王 涛 编著

四川大学出版社

责任编辑:毕 潜

责任校对:贾朝辉

封面设计:罗 光

责任印制:李 平

内 容 提 要

本书共 12 章,主要包括:C 语言程序设计基础,算法,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组与指针,函数与模块化程序设计方法,结构体和共用体,预处理命令与带参数的主函数,文件操作,C 语言图形介绍及 C++ 简介等。本书面向初学者,使略有计算机基础的人都能较容易地学会 C 语言编程。书中所采用的程序实例,充满趣味性和实用性,语言叙述通俗易懂,难点分散,概念清晰,层次分明。

本书可作为高等院校的计算机和非计算机专业的教科书和参考书。同时,也可供各行各业从事计算机工作的人员使用。

图书在版编目(CIP)数据

C 语言程序设计/顾治华,陈天煌,王涛编著. —成都:四川大学出版社, 2004.1

ISBN 7-5614-2746-8

I.C... II.①顾...②陈...③王... III.C 语言-程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2003) 第 122837 号

书 名 C 语言程序设计

编 著 顾治华 陈天煌 王 涛

出 版 四川大学出版社

地 址 成都市一环路南一段 24 号 (610065)

印 刷 华西医科大学印刷厂

发 行 四川大学出版社

开 本 787 mm×1 092 mm 1/16

印 张 15.75

字 数 380 千字

版 次 2004 年 1 月第 1 版

印 次 2004 年 1 月第 1 次印刷

印 数 0 001~3 000 册

定 价 28.00 元

◆读者邮购本书,请与本社发行科联系。电话:85408408/85401670/85408023 邮政编码:610065

◆本社图书如有印装质量问题,请寄回出版社调换。

◆网址:www.scupress.com.cn

版权所有◆侵权必究

前 言

“C 语言程序设计”作为计算机文化基础、技术基础和应用基础三个层次中第二层次的一门主要课程，是所有理工科高校学生的必修课，又是全国计算机等级考试（二级）的主要语言。C 语言作为目前广为流行的面向对象程序设计主要语言之一的 C++ 的核心，它已成为广大计算机应用人员和计算机爱好者、初学者的必修语言。

在多年的教学过程中，我们发现，相当多的学生学完了“C 语言程序设计”课程以后，仅仅了解和掌握了一些语句的语法知识和语义，但不会应用语言来编写程序，把编程视为十分艰难而又高不可攀的工作。

我们在介绍 C 语言的同时，注重讲解如何应用 C 语言来编程，试图帮助读者克服畏难情绪，在轻松、愉快的气氛中探索程序设计的奥妙。本书具有如下特点：

（1）面向初学者，使略有计算机基础的人都能较容易地学会 C 语言编程。书中所采用的程序实例，充满趣味性和实用性，语言叙述通俗易懂，难点分散，概念清晰，层次分明。

（2）为了帮助读者学会程序设计，我们还专门介绍了一些程序设计的常用方法，如穷举法、迭代法和递推法等，使读者在学习和掌握一门计算机语言的同时，养成良好的程序设计习惯。

（3）为了帮助读者能充分利用 Internet 丰富的学习资源，在书中附录部分提供了百余个中英文的关键词，由这些关键词可搜索到更多更新的 C 语言程序设计的文献资料。

本书在写作的过程中，与多名多年讲授该课程的教师进行过讨论，汲取了他们许多宝贵的教学经验，同时，还得到了院系领导的大力支持。在此，一并表示衷心的感谢。

由于编者水平有限，书中错误在所难免，恳请读者批评指正，并多多提出宝贵意见。

作者于武汉理工大学

2003 年 12 月

目 录

第1章 C语言程序设计基础	1
1.1 程序与程序设计	1
1.2 C语言简介	2
1.2.1 C语言的发展历史	2
1.2.2 C语言的特点	3
1.3 简单的C语言程序	4
1.4 C语言程序的上机步骤	5
1.4.1 Turbo C编译C语言程序	5
1.4.2 UNIX操作系统下运行C语言程序	6
1.4.3 在DOS中用Microsoft C 6.0编译程序运行C语言程序	6
1.5 C语言的基本词法	6
1.5.1 标识符	7
1.5.2 关键字	7
1.5.3 运算符	8
1.5.4 分隔符	8
1.5.5 常量	8
1.5.6 C语言的字符集	8
1.6 C语言程序的基本结构	8
1.6.1 顺序结构	8
1.6.2 分支结构	9
1.6.3 循环结构	9
习题1	10
第2章 程序的基石——算法	11
2.1 算法的概念	11
2.2 算法的设计与表达	12
2.2.1 基本的程序语句	12
2.2.2 算法的表示方法	13
2.3 简单的算法实例	14
2.4 结构化程序设计方法简介	17
习题2	19
第3章 顺序结构程序设计	20
3.1 基本数据类型	20
3.1.1 整型	20

3.1.2	实型.....	23
3.1.3	字符型.....	25
3.2	常用的运算符和表达式.....	28
3.2.1	赋值运算符.....	28
3.2.2	算术运算符.....	31
3.2.3	位运算符.....	33
3.2.4	条件运算符和逗号运算符.....	36
3.2.5	长度测试运算符.....	37
3.2.6	数值型数据的混合运算.....	37
3.3	表达式及赋值语句.....	38
3.3.1	表达式和语句.....	38
3.3.2	赋值语句.....	38
3.4	基本输入输出操作的实现.....	39
3.4.1	基本输入输出的概念.....	39
3.4.2	字符数据的输入输出.....	39
3.4.3	格式化输入输出函数.....	41
3.5	顺序结构程序设计实例.....	47
	习题 3.....	50
第 4 章	选择结构程序设计.....	53
4.1	关系运算符与关系表达式.....	53
4.1.1	关系运算符及其优先次序.....	53
4.1.2	关系表达式.....	54
4.2	逻辑运算符和逻辑表达式.....	54
4.2.1	逻辑运算符及其优先次序.....	54
4.2.2	逻辑表达式.....	55
4.3	if 语句.....	56
4.3.1	if 语句的三种形式.....	56
4.3.2	if 语句的嵌套.....	59
4.4	switch 语句.....	61
4.5	选择结构程序设计实例.....	65
	习题 4.....	68
第 5 章	循环结构程序设计.....	70
5.1	循环结构的应用场合.....	70
5.2	goto 语句.....	70
5.3	while 语句.....	71
5.4	do-while 语句.....	73
5.5	for 语句.....	75
5.6	多重循环.....	77
5.7	几种循环语句的比较.....	78

5.8 转移控制语句.....	79
5.8.1 break 语句.....	79
5.8.2 continue 语句.....	79
5.9 单重循环及多重循环结构的实例.....	80
5.10 常用程序设计方法——穷举法.....	83
习题 5.....	84
第 6 章 数组与指针.....	86
6.1 数 组.....	86
6.1.1 数组的概念.....	86
6.1.2 数组的定义、初始化和引用.....	87
6.1.3 字符数组.....	93
6.2 指针的概念.....	99
6.2.1 地址与指针的概念.....	100
6.2.2 指针的定义与引用.....	101
6.3 指针与数组.....	104
6.3.1 指向一维数组的指针.....	104
6.3.2 指向多维数组的指针.....	108
6.4 字符串的指针.....	111
6.5 指针数组和指向指针的指针.....	114
6.5.1 指针数组.....	114
6.5.2 指向指针的指针.....	116
6.6 常用程序设计方法——迭代法和递推法.....	117
6.6.1 迭代法.....	117
6.6.2 递推法.....	118
习题 6.....	120
第 7 章 函数与模块化程序设计方法.....	122
7.1 函 数.....	122
7.2 函数的定义.....	123
7.3 函数的参数和返回值.....	126
7.3.1 形参与实参.....	126
7.3.2 函数的返回值.....	127
7.4 函数的调用.....	128
7.4.1 函数调用的一般形式.....	128
7.4.2 库函数的调用.....	129
7.4.3 地址传送方式传递数据.....	129
7.5 函数的嵌套与递归调用.....	131
7.5.1 函数的嵌套调用.....	131
7.5.2 函数的递归调用.....	133
7.6 数组与函数参数.....	135

7.6.1 数组元素作为函数实参.....	135
7.6.2 数组名作为函数实参.....	135
7.7 指针与函数.....	140
7.7.1 指向函数的指针.....	140
7.7.2 返回指针的函数.....	143
7.8 变量的存储类别.....	144
7.8.1 变量的作用域.....	144
7.8.2 动态存储和静态存储.....	147
7.8.3 auto 变量.....	147
7.8.4 局部静态变量.....	148
7.8.5 寄存器变量.....	149
7.9 模块化程序设计方法.....	150
7.9.1 模块化程序设计方法的指导思想.....	150
7.9.2 模块分解的原则.....	150
习题 7.....	151
第 8 章 结构体和共用体.....	152
8.1 结构体类型的定义.....	152
8.2 结构体变量的定义和引用.....	153
8.2.1 结构体变量的定义.....	153
8.2.2 结构体变量的引用.....	155
8.3 结构体数组.....	156
8.3.1 结构体数组的定义.....	156
8.3.2 结构体数组的初始化.....	157
8.4 指向结构体类型数据的指针.....	158
8.4.1 指向结构体变量的指针.....	158
8.4.2 指向结构体数组的指针.....	159
8.4.3 指向结构体的指针作函数参数.....	160
*8.5 内存的动态分配.....	162
8.5.1 动态分配内存.....	162
8.5.2 链表概述.....	162
8.5.3 建立链表.....	163
8.5.4 链表的删除操作.....	166
8.5.5 链表的插入操作.....	167
8.6 共用体类型.....	169
8.6.1 共用体类型.....	169
8.6.2 共用体类型变量的引用方式.....	169
8.6.3 共用体类型变量的特点.....	169
8.7 枚举类型.....	171
8.8 用户自定义类型.....	173

习题 8.....	174
第 9 章 预处理命令与带参数的主函数.....	175
9.1 宏定义.....	175
9.1.1 不带参数的宏定义.....	175
9.1.2 带参数的宏定义.....	177
9.2 文件包含处理.....	179
9.3 条件编译.....	180
9.4 带参数的主函数.....	181
习题 9.....	182
第 10 章 文 件.....	183
10.1 C 文件概述.....	183
10.2 文件的打开与关闭函数.....	184
10.2.1 文件指针.....	184
10.2.2 打开文件.....	185
10.2.3 关闭文件.....	186
10.3 文件的读/写函数.....	186
10.3.1 fputc () 函数和 fgetc () 函数.....	186
10.3.2 fgets () 函数和 fputs () 函数.....	188
10.3.3 fprintf () 函数和 fscanf () 函数.....	188
10.3.4 fread () 函数和 fwrite () 函数.....	189
10.4 文件处理的其他常用函数.....	190
10.4.1 文件的定位.....	190
10.4.2 出错检测.....	193
习题 10.....	193
*第 11 章 C 语言图形介绍.....	194
11.1 随机数的使用.....	194
11.2 键盘的控制.....	194
11.3 鼠标控制.....	195
11.4 光标控制.....	197
*11.5 Turbo C 的图形功能介绍.....	198
11.5.1 图形模式的初始化.....	198
11.5.2 独立图形运行程序的建立.....	200
11.5.3 屏幕颜色的设置和清屏函数.....	201
11.6 图形方式下的文本输出函数.....	202
11.6.1 文本输出函数.....	202
11.6.2 定义文本字型函数.....	203
11.6.3 文本输出字符串函数.....	205
11.7 常用统计图形的绘制.....	206
11.8 综合应用实例.....	211

第 12 章 C++简介	213
12.1 C++的特点.....	213
12.1.1 C 转入 C++时不需改变的内容	213
12.1.2 C 转入 C++时的一些与类无关的新特性	214
12.2 C++的核心新特性——类.....	217
12.2.1 类和对象.....	217
12.2.2 类成员的访问.....	218
12.2.3 构造函数和析构函数.....	219
附 录.....	223
附录 I 常用字符与 ASCII 代码对照表	223
附录 II C 语言中的关键字	225
附录 III 运算符和结合性.....	226
附录 IV Turbo C2.0 上机操作指南.....	228
附录 V C 语言常用库函数	233
附录 VI 常用词汇中英文对照表.....	239
参考文献.....	242

第1章 C语言程序设计基础

1.1 程序与程序设计

计算机通过执行程序完成其工作，如计算、控制、文字处理、图形处理、网络通信等。所谓程序，就是一组指令和数据的集合。计算机可以执行的程序通常以文件的形式存放在磁盘上，当需要执行时调入内存。

可执行文件包含的主要部分是一些二进制的机器指令和必需的数据。机器指令直接控制计算机的每一个部件（如寄存器、存储器单元），机器指令的表达方式称为机器语言。机器语言与特定的计算机有关，它能被特定的计算机所识别，但不利于人的理解和编程。

为了便于理解，人们开始用助记符代替机器指令，这便是汇编语言。如 `ADD AX BX`，把寄存器 `AX` 与 `BX` 中的值相加，结果放到 `AX` 中。汇编语言只是用助记符代替二进制机器指令，所以它是面向机器的。

机器语言和汇编语言与机器硬件有关，属于“低级语言”。在程序设计中，一般采用人们易于理解的类自然语言，这些语言称为“高级语言”，如 `C`，`PASCAL`，`BASIC`，`FORTRAN` 等。高级语言表达了人控制计算机的意图，例如：

```
printf("hello world!\n");          /*在屏幕上输出一串字符*/
```

`CPU` 只能理解机器语言，不能理解汇编语言和高级语言，必须把汇编语言或者高级语言编写的程序“翻译”成机器语言才能执行。汇编语言“翻译”成机器语言的过程称为“汇编”。高级语言“翻译”成机器语言有两种方式：一种是“解释”，一种是“编译”。`C` 语言属于编译型语言。编译的原理就是由编译程序把源程序编译、连接成可执行文件，然后由机器直接执行。`C` 语言源程序的运行过程如图 1-1 所示。

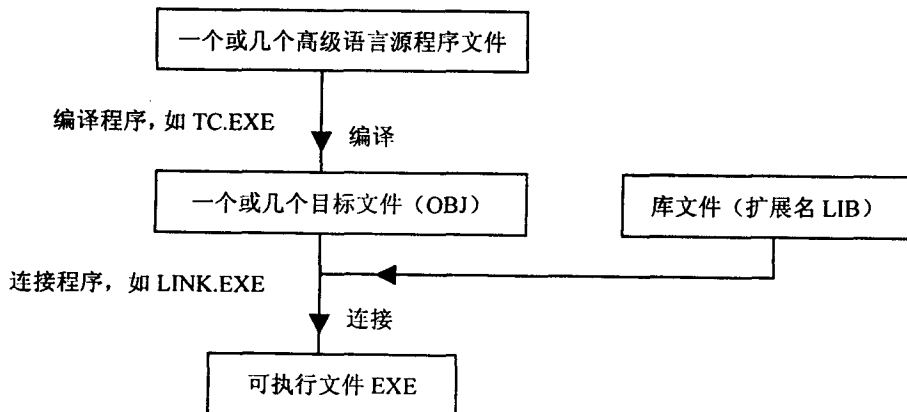


图 1-1 C 语言源程序的运行过程

程序员编写的源程序必须遵循编译程序规定的语法。语法是编写程序的规则，各种类型的语法规则产生了 C, PASCAL, BASIC, FORTRAN 等语言。C 语言语法标准如下：

(1) ANSI C: 1983 年由美国国家标准化协会 (ANSI) 制定。

(2) 87 ANSI C: 1987 年由美国国家标准化协会制定。

程序设计所需的软件条件 (称为开发环境) 如图 1-2 所示。

源程序编辑程序 (Editor)	集成开发环境 IDE (Integrated Developer Environment)
编译程序 (Compiler)	
连接程序 (Linker)	
调试程序 (Debugger)	

图 1-2 C 语言开发环境

程序开发环境的核心是编译程序，它把程序员编写的类自然语言源程序翻译为机器指令，同时，从应用的角度讲，它提供了程序设计的思想。主要的程序设计思想包括结构化程序设计思想和面向对象程序设计思想。典型的程序开发环境及其特点见表 1-1。

表 1-1 C 语言开发环境及其特点

Microsoft	Borland (Inspires)	特 点
MSC	Turbo C	结构化程序设计 开发 DOS 程序
Visual C++	C++ Builder	面向对象程序设计 可视化程序设计环境 专业化 Windows 9X/NT 设计
Visual Basic	Delphi (类 PASCAL 语法)	面向对象程序设计 可视化程序设计环境 快速设计 Windows 9X/NT 程序

程序设计的过程如下：

- (1) 问题的提出，要达到的要求。
- (2) 确定数据结构和所采用的算法。
- (3) 编制程序。
- (4) 调试程序。
- (5) 整理并写出文档。

1.2 C 语言简介

1.2.1 C 语言的发展历史

C 语言是现在很流行的一种高级程序设计语言。它既可以用来写应用软件，也可以用来写系统软件。

以前的操作系统都是用汇编语言编写的,但是由于汇编语言依赖于特定的计算机硬件,可移植性很差,而且汇编语言编写的程序都比较难以读懂。于是人们就想用一种高级语言来编写系统软件,但在高级语言中却很难实现汇编程序对硬件的操作能力。C语言正是在这种背景下被设计出来的。

C语言是随着UNIX操作系统的开发而诞生的,是在B语言的基础上发展起来的。在此之前,人们设计了几种高级语言,但是离硬件都比较远,无法用来写系统软件。1970年,贝尔实验室的Ken Thompson在其他语言的基础上,设计了很简单又很接近硬件的B语言,并且用它编写了UNIX操作系统。1972年至1973年,贝尔实验室的D.M.Ritchie在B语言的基础上设计出了C语言,并与K.Thompson合作用C语言改写了UNIX操作系统。随着UNIX的广泛应用,C语言也得到了长足的推广。现在,C语言已成为全世界应用最广泛的几种计算机语言之一。

1.2.2 C语言的特点

1.2.2.1 C语言的优点

人们之所以要采用C语言编制各种各样的程序,是因为C语言具有如下一些独特的优点:

(1) C语言简洁,灵活。C语言程序格式书写自由,不像FORTRAN那样有严格的格式。与PASCAL相比,C语言的关键字简练,源程序短,输入的工作量比较小。采用C语言编程,可以使程序员专注于算法设计上,不必过多地考虑格式的限制。

(2) C语言有丰富的运算符,使源程序精练,生成的代码质量高,运行速度快。

(3) 数据类型丰富,能实现各种复杂的运算,尤其是指针类型数据,能使程序更加灵活和多样。

(4) 语法限制不是很严格。例如C语言对数组下标越界不做检查,由程序员来保证程序的正确性。同时,对变量类型的使用比较灵活,例如,整型与字符型及逻辑型数据可以互相通用。

(5) C语言可以直接访问物理地址和计算机硬件,能进行位操作,可以实现汇编语言的很多功能。因此,C语言具有高级语言和低级语言的双重功能,可以用来编写系统软件。

(6) 用C语言写的程序可移植性好,一般不做修改或者做少量的修改就能用于不同的计算机和不同的操作系统。

读者在编写C语言程序的过程中,特别是在与其他语言对比时,才会对以上的特点有更加深入的体会。

1.2.2.2 C语言程序的结构

(1) C语言是结构化的程序设计语言。

(2) C语言是模块化的程序设计语言,程序由许多函数组成。C语言所编制的程序必须有一个称为main()的主函数,而且只能有一个主函数,“{”和“}”分别表示函数的起点和终点,相当于PASCAL的BEGIN……END。函数之间可以相互调用、递归调用,但一般函数不能调用主函数。

(3) C语言的程序可以调用其他文件的函数,因此,一个C语言程序可以由许多文

件组成，便于合作开发。

(4) C 语言的一个语句可以放在一行，也可以放在多行；C 语言程序的一行也可以放多个语句。C 语言的语句都要用 “;” 作为结束标志。为了便于阅读，编写 C 语言的程序应遵循一定的规则，例如，嵌套循环时应该有缩行。

(5) 为便于 C 语言程序的维护和帮助人们的阅读，C 语言的关键语句应该有注释，注释部分必须用 “/*” 和 “*/” 括起来，并且 “/” 和 “*” 之间不能有空格；否则，编译程序在编译时会忽略掉 “/*” 和 “*/” 之间的内容。

(6) C 语言的程序一般要有头文件。头文件在程序的开始用 “#include” 做出说明；头文件中可以是对程序中所用变量的说明，也可以是引用的库函数。

(7) C 语言区分大小写，因此在使用 C 语言时应特别注意。

(8) C 语言的程序总是从主函数开始执行，并且终止于主函数。

总之，C 语言灵活性大，功能强，可以编写出各种类型的程序。程序员使用 C 语言编程感到限制少，可以自由地编程。但是，从学习语言的角度来说，学习 C 语言比学习其他高级程序设计语言要难一些。学习 C 语言的困难主要来自于 C 语言的灵活性，其实，只要掌握了 C 语言的基本语法规则，多上机练习，C 语言是可以学好的。

1.3 简单的 C 语言程序

我们以 “Hello World!” 作为本书的开始程序。

【例 1-1】 最简单的 C 语言程序。

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

程序执行后在屏幕上输出：

Hello World!

从这个程序可以看出，无论 C 语言程序是简单还是复杂，都要有一个 main() 函数，它是程序执行的第一个函数，即程序的入口点。printf() 是一个输出函数，要调用这个函数，程序的头文件必须包含 <stdio.h> (C 语言把 printf 视为标准输出函数，可省略 #include "stdio.h")。

【例 1-2】 两个数求和。

```
#include "stdio.h"
main()
{
    int a,b,sum;
    a=3;
    b=6;
    sum=a+b;
```

```
printf("sum=%d\n",sum);
```

```
}
```

程序的输出结果为:

```
sum=9
```

程序先定义三个变量,然后分别给变量 a 和变量 b 赋值,变量 sum 是变量 a 和变量 b 的和,程序输出的结果正是 a 和 b 的和。“%d”表示输出的是十进制整数。

1.4 C语言程序的上机步骤

C语言是编译型语言,源程序必须经过编译才能在计算机上执行。

1.4.1 Turbo C 编译 C语言程序

在 Turbo C 所在的目录下,直接键入 TC,就可打开 C 程序的编辑环境,如图 1-3 所示。

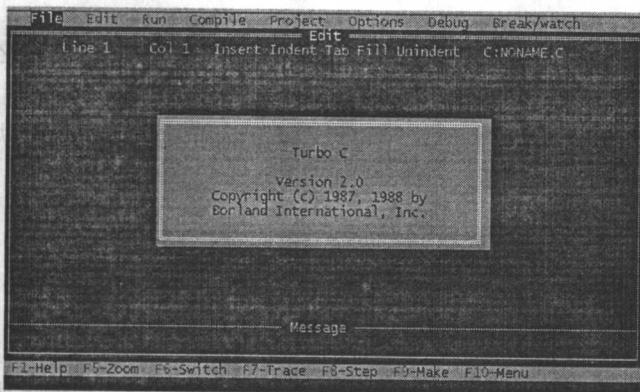


图 1-3 C语言程序的编辑环境

在 Turbo C 的集成环境中,有一行主菜单,下面简单介绍这些菜单。

File 菜单用来对文件进行操作,包括装载文件、建立新文件、存储文件等。按 Alt+F 组合键可以下拉 File 菜单,如图 1-4 所示。

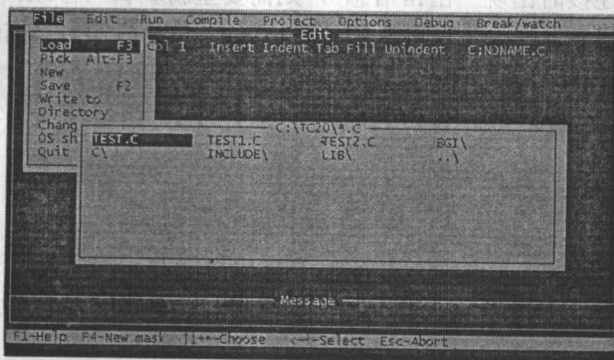


图 1-4 File 菜单

选择“New”，建立一个新文件，然后集成环境就打开编辑窗口，读者可以在此输入 C 语言源程序。

读者编辑完源程序后，要对源程序进行编译，生成可执行文件，方可执行。按下 Alt+C 组合键，选择“Compile to OBJ”，生成目标文件，然后选择“Link EXE file”连接目标文件，也可以直接选择“Make EXE file”生成可执行文件。如果源程序没有语法和语义错误，则生成后缀为“.exe”的可执行文件。如果源程序存在错误，集成环境会指出错误的行，读者可以打开编辑窗口修改源程序。

生成可执行文件后，按下 Alt+R 组合键，选择“Run”，或者直接按下 Ctrl+F9，可以执行此文件。或者先退出集成环境，然后在可执行文件所在目录下，直接输入可执行文件名，也可以执行此文件。

1.4.2 UNIX 操作系统下运行 C 语言程序

在 UNIX 中打开屏幕编辑程序 vi，将源程序输入。然后对源程序进行编译，直接输入命令：

`cc 文件名.c`

在 UNIX 中，连接也是由 cc 完成的，最后得到可执行文件，文件名由系统自动确定为 a.out。

如果不想用系统指定的文件名，也可以在编译时指定所生成的可执行文件名。输入以下命令：

`cc -o 可执行文件名.out 文件名.c`

直接输入文件名，程序即可执行。

1.4.3 在 DOS 中用 Microsoft C 6.0 编译程序运行 C 语言程序

(1) 编译 C 语言源程序可以用任何全屏幕编辑系统输入和编译源程序。假设已输入和编辑好的源文件名为 al.c。

(2) MS C 提供了一个功能很强的 CL 命令，可一次性完成程序的编译和连接。CL 是 Compile 和 Link 的第一个字母。用法如下：

`CL al.c`（对源程序 al.c 进行编译和连接）

如果在编译和连接时出现“出错信息”，则需要重新编辑（修改）源程序。

编译和连接完成后，产生一个可执行文件 al.exe。执行程序，只需输入可执行文件名：al，即可得到运行结果。

以上步骤只需上机试一下，便可明白。

1.5 C 语言的基本词法

在 C 语言中使用的词类分为六类，即标识符、关键字、运算符、分隔符、常量和注释符。

1.5.1 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外,其余的都由用户自己定义。C 语言规定,标识符只能是字母(A~Z, a~z)、数字(0~9)、下划线(_)组成的字符串,并且其第一个字符必须是字母或下划线。例如:

合法的标识符有: a, x, x3, BOOK_1, sum5, _x7。

非法的标识符有: 3s(以数字开头), s*T(出现非法字符*), -3x(以减号开头), bowy-1(出现非法字符-)。

在使用标识符时还必须注意以下几点:

(1) 标准 C 不限制标识符的长度,但它受各种版本的 C 语言编译系统限制,同时也受到具体机器的限制。例如,在某版本 C 中规定标识符前 8 位有效,当两个标识符前 8 位相同时,则被认为是同一个标识符。Turbo C 规定标识符的长度为 32。在编写程序时,应对系统所规定的标识符的长度有所了解,以免造成不必要的错误。这种错误不会被编译系统发现,所以应特别小心。一般情况下,标识符的长度不要超过 8 个字符。

(2) 在标识符中,大小写是有区别的。例如, SUM 和 sum 是两个不同的标识符。变量名应尽量使用小写字母,以增加程序的可读性。

(3) 标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号。因此,命名应尽量有相应的意义,以便于阅读理解,做到“顾名思义”。一般不应用简单的无意义的符号作为变量名,如 a, b 等。

在 C 语言中,所有的变量都是先定义后应用,使用没有定义的变量名会被认为是“非法”的。

1.5.2 关键字

关键字是由 C 语言规定的具有特定意义的字符串,通常也称为保留字。用户定义的标识符不应该与关键字相同。C 语言的关键字分为下面三类:

(1) 类型说明符。

类型说明符用于定义和说明变量、函数或其他数据结构的类型,如 int, double, float, long, short, auto, signed, static, struct, unsigned, char, enum, extern, register, union 等。

(2) 语句定义符。

语句定义符用于表示一个语句的功能,如条件语句的语句定义符 if-else, 循环语句的语句定义符 do, while, for, goto 等。

(3) 预处理命令字。

预处理命令字用于表示一个预处理命令,使用时前面要加“#”,如 include, define, ifdef, ifndef, undef, endif 等。

关键字后必须有空格、圆括号、尖括号、双引号等分隔符,否则关键字会与其他字符一起组成新的标识符。