C语言程序设计

周学毛 易 卫 周中柱/编著

天津大学出版社
TIANJIN UNIVERSITY PRESS

C语言

程序设计

周学毛　易　卫　周中柱　编著

天津大学出版社
TIANJIN UNIVERSITY PRESS

内 容 简 介

本书以C语言程序设计为主线，采用任务驱动、问题牵引、案例支撑的编写模式，全面介绍了C语言程序的基本语法和结构。全书共13章，具体内容包括：C语言基础、基本数据类型、表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、函数、数组类型、结构体类型与共用体类型、指针类型、文件类型、编译预处理和图形处理、C语言程序设计实例。

本书适合作为计算机及相关专业“C语言程序设计”课程的专业教材，也可作为C语言程序设计的培训教材和自学用书。

图书在版编目（CIP）数据

C语言程序设计/周学毛，易卫，周中柱编著. —天津：
天津大学出版社，2013.6

ISBN 978-7-5618-4725-1

Ⅰ. ①C… Ⅱ. ①周… ②易… ③周… Ⅲ. ①C语言—程序设计
Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2013）第137764号

出版发行	天津大学出版社
出 版 人	杨欢
地　　址	天津市卫津路92号天津大学内（邮编：300072）
电　　话	发行部：022-27403647
网　　址	publish.tju.edu.cn
印　　刷	河北省昌黎县思锐印刷有限责任公司
经　　销	全国各地新华书店
开　　本	185mm×260mm
印　　张	16.5
字　　数	412千
版　　次	2013年7月第1版
印　　次	2013年7月第1次
定　　价	33.00元

前　言

本书的编写以作者前些年编著的《新编C语言程序设计教程》为基础，融入近几年C语言与程序设计的教学研究和教改实践成果，基于看得懂、学得会、用得上的编写理念，遵循任务驱动、问题牵引、案例支撑的编写模式，可以让读者快速自主地进入C语言程序设计世界。

本书继续保留了原教材的特色，依然以程序设计为中心，强调学C语言是学程序设计的思想、学程序设计的方法、学程序设计的技术，强调有问题、有方法、有实现。学程序设计语言只是手段，学程序设计方法、学计算机解决问题的方法才是目的。

本书还对原有体系架构做了部分调整，除将原第三部分析出将单独编写成《C++速成教程》，保持了原入门、进阶两个部分已得到读者高度认可的章节与内容，稍后将推出配套的《C语言程序设计学习指导》一书。

全书共计13章，第1～7章为C语言的入门部分，是C语言程序设计的基础篇、重点篇，详细介绍了C语言的语法成分、C语言程序的结构、基本数据类型与数据描述方法、C语言的基本语句结构、函数。通过这部分内容的介绍，可以让读者熟练掌握C语言的语法，认识C语言程序的结构，了解结构化程序设计的方法，能动手编制与调试简单的C语言程序。

第8～13章为进阶部分，是C语言程序设计的关键篇、中心篇，主要对数据类型进行扩充，对数据描述方法进行拓展，全面介绍数组、结构体、指针及文件等用户自定义类型的使用，同时介绍一些常用的算法及编程技巧，通过大量的典型实例让读者基本掌握结构化程序设计方法，能编制一般的C语言程序。第13章介绍了四个C语言程序设计的综合实例，可供读者进行综合性训练与实践。希望通过这部分内容的介绍，让读者进一步掌握C语言程序设计方法，提高C语言程序设计能力。

对于初学程序设计的读者，建议一定要多读程序，多动手编程序，边学边做，边做边学，小题大作，在实践中掌握基础知识。程序设计的学习强调先模仿，在模仿的基础上改进，在改进的基础上提高。教学中可以详讲第1～8章，第9～13章则由学生自主学习。

由于作者水平有限，加之时间仓促，书中错漏与不足之处在所难免，希望广大读者提出宝贵意见和建议，在此表示感谢！

另外，还要特别感谢天津大学出版社在本书修编过程中给予的大力支持！

作者于岳麓山

2013年1月

目　录

第 1 章

C 语言基础

程序设计语言是程序设计人员与计算机进行对话的语言，它遵循一定的规则和形式，构成程序的开发平台。为满足计算机的各种应用，人们设计了各种程序设计语言。

C 语言是一种功能通用、应用广泛的高级语言，既可用来开发系统程序，又可用来开发各种应用程序。

本章作为开篇，将介绍 C 语言的特点、C 语言的基本语法成分、C 程序的结构及 C 程序的实现。

1.1 C 语言概述

1.1.1 C 语言的产生、发展与标准

1．C 语言的产生

C 语言于 1972—1973 年，诞生于美国的贝尔实验室，由 D.M.Ritchie 创建。

C 语言的产生基于两个方面的需要：一是为满足 UNIX 操作系统开发的需要，UNIX 操作系统是一个通用的、复杂的计算机管理系统；二是为拉近高级语言与硬件之间距离的需要，寻找一种语言，能够集高级语言与汇编语言优点于一身。

C 语言面对实际应用的需要而产生，直至今日仍不改初衷。

2．C 语言的发展

对 C 语言追本求源，可追溯到 ALGOL 语言。1960 年出现的 ALGOL 语言是一种面向问题的高级语言，但其远离硬件，不适于开发系统软件。1963 年，英国剑桥大学推出 CPL 语言，CPL 语言比 ALGOL 语言更接近硬件一些，但规模较大，难以实现。1969 年，剑桥大学的 Martin Richards 对 CPL 语言进行简化，推出 BCPL 语言。1970 年，贝尔实验室的 Ken Thompson 为 DEC 公司 PDP-7 计算机上运行的一种早期 UNIX 操作系统设计了一种类 BCPL 语言，称为 B 语言。B 语言规模小，接近硬件，1971 年在 PDP-11 计算机上实现。BCPL 语言和 B 语言都是无类型的语言，过于简单，功能有限。

1972—1973 年，贝尔实验室的 D.M.Ritchie 在保留 B 语言优点的基础上，创建了 C 语言。1973 年 UNIX 操作系统被用 C 语言改写，称为 UNIX 第 5 版。最初的 C 语言只是一种 UNIX 操作系统的工作语言，依附于 UNIX 系统，主要在贝尔实验室内部使用。UNIX 以后的第 6 版、第 7 版、SYSTEM Ⅲ和 SYSTEM Ⅴ都是在第 5 版的基础上发展起来的，C 语言也作了多次改进。1975 年，UNIX 第 6 版的公布，使 C 语言受到人们的普遍关注。

UNIX 操作系统的广泛使用，促进了 C 语言的迅速发展与普及，也促进了 UNIX 操作系统的推广。1978 年，独立于 UNIX 操作系统和 PDP 计算机的 C 语言出现了，从此 C 语言被迅速移植到大、中、小型与微型机上。当年，B.W.Kernighan 和 D.M.Ritchie 以 UNIX 第 7 版的 C 编译程序为基础，出版了影响深远的名著《The C Programming Language》。

目前，C 语言已经发展成为迄今为止最流行的计算机程序设计语言。C++语言是 C 语言发展的新阶段，是一种更好的 C 语言，是应用广泛的面向对象的程序设计语言。

3．C 语言的标准

Kernighan 和 Ritchie 的名著《The C Programming Language》出版不久，各种 C 语言编译系统纷纷出现，由于这本书的存在，使得绝大多数 C 语言保持着高度的兼容性，这本书描述的 C 语言成为 C 语言世界的当然标准，简称 K&R C，给出了 C 语言的经典定义。

然而，C 语言的广泛应用还是在不同的开发团体之间出现了问题，各机构推出了自己的 C 语言与版本，某些执行过程的微小差别却不时引起 C 程序之间的不兼容。美国国家标准协会（ANSI）从 1983 年开始经过长达五年的努力，制定了 C 语言的新标准—— ANSI C，现在提及 C 语言的标准是指新标准。ANSI C 比原标准 C 有很大的发展，解决了经典定义中的二义性，给出了 C 语言的新特点，Kernighan 和 Ritchie 也以新标准改写了他们的经典著作。任何 C 程序都必须遵循 ANSI C 标准，本教材的主体也以 ANSI C 作为基础。

1.1.2 C 语言的特点

C 语言是一种通用、灵活、结构化、标准化、使用普遍的编程语言，它能完成用户想完成的任何任务，特别适合于进行系统程序设计和对硬件进行操作的场合。

C 语言本身不对程序员施加过多限制，因而成为专业程序员优先选择的基础语言。

C 语言的主要特点如下。

（1）C 语言简洁、紧凑，压缩了一切不必要的成分。ANSI C 有 32 个关键字，Turbo C 有 58 个关键字，9 种控制语句，书写形式自由。

（2）C 语言运算丰富，将括号、赋值、强制类型转换、取变量地址等都以运算实现。ANSI C 提供 34 种运算，Turbo C 提供 44 种运算，灵活使用这些运算可以实现其他高级语言难以实现的操作。C 语言的表达式简练、多样、灵活、实用，其表达式加上分号就可以构成语句。

（3）C 语言数据类型丰富，具有现代语言的各种数据类型。用户还能够扩充数据类型，实现各种复杂的数据结构，完成各种问题的数据描述。尤其是 C 语言的指针类型非常有特色，可指向各种数据，完成各种数据的高效处理。C 语言对数据不但作类型上的描述，还提供存储属性考虑。

（4）C 语言是一种结构化程序设计语言，具有结构化语言所要求的基本结构。C 语言用函数作为结构化程序设计的实现工具，实现程序的模块化。C 程序由若干程序文件组成，一个程序文件由若干函数构成。

（5）C 语言是高级语言中的低级语言。C 语言允许直接访问物理地址，能进行位运算，实现汇编语言的大部分功能，且直接对硬件进行操作。

（6）C 语言提供编译预处理机制，有利于 C 程序的编写和调试。

（7）C 语言编译系统小，生成目标代码质量高，程序执行效率高。

（8）C 语言输入和输出功能用库函数实现，编写的程序移植性好。

（9）C 语言语法限制不太严格，程序设计自由度大，对程序员要求较高。

1.2 基本语法成分

本节介绍 C 语言的字符集、关键字及标识符等基本语法成分。

1.2.1 字符集

字符是可以区分的最小符号，字符集构成语言与程序的原始基础。C 语言字符集是 ASCII 字符集的一个子集，包括英文字母、数字及特殊字符。

- 英文字母：a～z 和 A～Z 。
- 数字：0～9。
- 特殊字符：空格、!、#、%、^、&、*、-、_、+、=、～、<、>、/、\、|、.、、，、：、；、？、"、"、(、)、[、]、{、}。

由字符集中的字符可以构成 C 语言进一步的语法成分，如标识符、关键字、特殊的运算符等。

1.2.2 标识符

标识符在语言与程序中用来标识各种语言与程序成分，用来命名语言与程序中的一些实体，如变量、常量、函数、类型、标号等的名字。

C 语言规定，标识符必须以字母或下划线开头，是字母、数字、下划线的序列。

以下是合法的标识符：

i，j，k，x，c，a1，a2，op，y_1，zhou_prg，radius，prime，program，sort，max，min，prg_1，cout，sun，day

以下是不合法的标识符：

a.1，1computer，x+y，! abc，99999，$100，π，3c

C 语言中标识符严格区分字母的大小写，标识符 abc 与标识符 ABC 是不同的标识符。习惯上，符号常量用大写字母表示，变量名称用小写字母表示。

标准 C 中，标识符的长度可以任意，一般有效长度为 8 个字符，Turbo C 中标识符最大长度为 32 个字符。

标识符取名时不能与关键字同名，也不要与系统预先定义的标准标识符（如标准函数）同名。系统使用的一些内部标识符往往以下划线开头，为避免与系统使用的内部标识符发生冲突，用户定义的标识符一般也不要用下划线开头。

标识符的名称一般选用相应的英文单词或拼音形式缩写，尽量不要使用简单代数符号，如 a、b、c、x、y、z 等。最好能“见名知义”，以提高程序的可读性。标识符大多数采用“匈牙利”表示法，每个单词第一个字母大写，如 StudentName、YearMonthDate。

由系统预先定义的标识符称为标准标识符，由用户定义的标识符称为自定义标识符，

程序设计中往往需根据实际需要定义大量的标识符。标识符必须先定义，后使用。

1.2.3 关键字

关键字又称为保留字，由系统提供，用以表示特定的语法成分，是构成 C 语言的语法基础。实际上，关键字是单列的一些非常特殊的有特定含义的准标识符。

如表示标准类型名称的 int、float 等关键字，定义类型的 array、struct 等关键字，表示语句特征的 if、switch、for、while 等关键字，表示运算符号的 sizeof 等关键字。

常见关键字有 32 个，参见附录 A。这些关键字将逐步引入，但希望读者能提前予以特别关注。

关键字有特定的语法含义，绝对不允许用户重新定义关键字，在程序中也绝对不能拼错。

1.2.4 运算符

运算符实际上可以认为是系统定义的函数名，这些函数作用于运算对象，得到一个结果值。运算符通常由 1 个或多个字符构成，C 语言沿用了大量的常规运算符。如加法运算符“+”、减法运算符“-”、地址运算符“&”、大于运算符“>”、不等运算符“！=”、逻辑与运算符“&&”、条件运算符“？:”、点运算符“.”、字节数运算符“sizeof”等。

根据运算对象的个数不同，可分为单目运算符、双目运算符和三目运算符，又称为一元运算符、二元运算符和三元运算符。

C 语言的运算符非常丰富，这些将在介绍数据类型与表达式的章节中予以详述，并请读者参考附录 B。

1.3 C 语言程序的结构

了解并掌握 C 语言程序的结构，对于 C 语言程序设计至关重要，是我们编写程序的模板。在讨论 C 语言程序结构之前，先看几个例子，以取得对 C 语言程序结构的一些基本认识。

例 1-1 在屏幕上输出“C 语言程序设计”。

```
/*程序 1-1，输出“C 语言程序设计”*/
#include "stdio.h"
main ( )
{ printf ("C 语言程序设计");
}
```

运行结果：C 语言程序设计

例 1-2 求两个整数之和。

```
/*程序 1-2，求两个整数之和*/
#include   "stdio.h"
main ( )
{ int   a, b;
  int   sum;
  printf ("请输入两个整数 a, b:");
```

```
    scanf ("%d, %d", &a, &b);
    sum=a+b;
    printf ("和=%d\n", sum);
}
```

输入数据：1999，5

运行结果：和=2004

例 1-3 求两个数的最大值。

```
/*程序 1-3，求两个数的最大值*/
#include "stdio.h"
float   max (x, y)   /*求最大值函数*/
float x, y;
{ float   t;
  if   (x>y)   t=x;
  else   t=y;
  return (t);
}
main ( ) /*主函数*/
{ float a, b;
  float m;
  printf ("请输入两个数:");
  scanf ("%f, %f", &a, &b);
  m=max (a, b);/*调用求最大值的函数*/
  printf ("最大值=%6.2f \n", m);
}
```

输入数据：12.1，3.45

运行结果：最大值=12.10

对本节所给例子，暂不必急于了解程序细节，而要先立足于从宏观上了解 C 程序的结构，理解 C 程序的组成，并尽量试着上机运行这些程序。

1.3.1 C 程序的结构

C 程序的核心成份是函数，C 程序的基本单位是语句。

1．函数

（1）C 程序由若干函数组成。

① 必须有一个且只能有一个主函数 main( )，主函数的名字为 main。

② 可以是系统预定义的标准函数，如 scanf 函数、printf 函数等，参见附录 C。

③ 大多数函数由程序员根据实际问题的需要进行定义，函数之间是平行的关系。

基于此，C 语言被称为函数式语言。一般地讲，C 程序由若干程序文件组成，每个程序文件由若干函数组成。

（2）函数由函数头与函数体两部分组成。

① 函数头是函数的说明部分，给出函数的特征描述，包括函数的属性、类型、名字、参数及参数类型。

如例 1-3 中求最大值函数的函数头：

```
float   max (x, y)
```

```
float    x, y;
/*函数属性为缺省，函数类型为实型，函数名字为 max，函数参数为 x、y，实型*/
```

② 函数体给出函数功能实现的数据描述和操作描述，是程序中用花括号括起的若干语句。

如例 1-1 中的函数体：

```
{ printf ("C 语言程序设计");
}
```

例 1-3 中求两个数的最大值函数 max 的函数体：

```
{ float  t;
  if  (x>y)  t=x;
  else  t=y;
  return (t);
}
```

函数体可以为空，可以没有数据描述部分。

由此可以给出理论上最小的 C 程序：

```
main ( )
{
}
```

2．语句

（1）语句是组成程序的基本单位，函数功能的实现由若干条语句完成。说明性语句完成数据描述，执行性语句完成操作描述。

（2）语句由若干关键字加以标识，如 if-else 语句、do-while 语句等。

（3）C 语言本身没有输入/输出语句，C 语言的输入/输出操作由 scanf 函数和 printf 函数等标准函数完成。

（4）C 语言语句必须以分号结束。

3．其他

1）预处理指令

C 程序开头往往包含以“#”开头的指令，它们是编译预处理指令。编译预处理指令在编译前处理，用以提高编译效率。如例 1-1 等程序中的#include"stdio.h"指令，用以指明包含文件。请参阅第 12 章。

2）程序注释

在程序中还有以/*开始、以*/结束的内容，它们是程序中的注释部分，用以帮助阅读程序。

注释对程序的执行没有任何影响，编译时注释将被过滤掉。加必要的注释只为增加程序的可读性，程序的可读性是衡量程序质量的重要指标。

1.3.2 C 语言程序的书写

C 语言程序的书写格式自由。采用一定的书写格式是人为的，主要用于增加程序的可读性。程序员应注意遵守这样的约定，以养成良好的程序设计风格。

（1）一般一行写一条语句。当然一行可以写多条语句，一条语句也可以写在多行上。

（2）整个程序采用紧缩格式书写。表示同一层次的相关功能的语句行对齐，缩进同

样多的字符位置。

如循环体中的语句要缩进对齐，选择体中的语句也要缩进对齐。一般缩进2个字符位置。

（3）花括号的书写方法较多，本书采用花括号对齐的书写方式。左边花括号处于第一条语句的开始位置，右边花括号独占一行，与左边花括号对齐。

（4）在程序中恰当地使用空行分隔程序中的语句块，增加程序的可读性。

1.4 C语言程序的实现

1.4.1 实现步骤

C语言程序在计算机上的实现与其他高级语言程序的实现一样，要经过编辑、编译、连接、运行四个步骤，在机器上实现前要进行数据分析与算法分析，如图1-1所示。

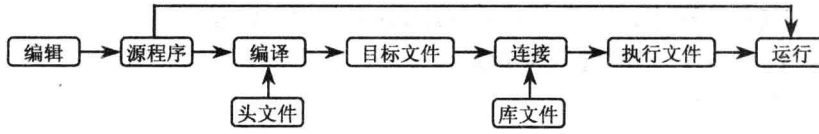

图1-1 C程序实现步骤

1．编辑

编辑是将C语言的源程序输入计算机，并以文本文件的形式存储在磁盘上。编辑是建立或修改C源程序文件的过程，源程序文件的扩展名为.C或.CPP。

编辑可以用任何文字处理软件完成，一般用编译器本身集成的编辑器进行编辑。

2．编译

C语言是以编译方式实现的高级语言，C语言程序的实现必须经过编译程序对源程序进行编译，生成目标代码程序，目标代码程序的扩展名为.OBJ。

编译前先进行预处理，编译过程主要进行词法分析和语法分析。

编译有错时，编译系统会列出错误的位置和种类。此时，要返回到编辑步骤修改源程序。修改后，再进行编译。

3．连接

编译形成的目标代码程序可被机器识别，但还不能直接执行，还需要将库文件与目标代码程序进行连接处理，连接工作由连接程序完成。经过连接，生成可执行文件，可执行文件的扩展名为.EXE。

如果连接有错，同样需返回到编辑步骤修改源程序。修改后，再进行编译、连接。

4．运行

一个C语言源程序经过编译、连接后生成了可执行文件。运行可执行文件，可通过编译系统下的运行功能来完成，也可像命令文件一样通过文件名执行。

程序运行后，一般在屏幕上显示运行结果。根据运行结果可以判断程序是否还有算法方面的错误。

编译时产生的错误是语法错误，运行时出现的错误是逻辑错误。出现逻辑错误时需要修改算法，重新编辑、编译、连接、运行程序。

1.4.2 程序的执行

C 程序的执行总是从主函数 main()开始（不管主函数在程序中什么位置）。其他函数通过被调用来执行。函数中的语句依语句的先后顺序执行，除非改变了程序的执行流程。

1.4.3 Turbo C 上机操作

Turbo C 是最简单的 C 程序上机操作环境，提供了命令行编译器与集成环境编译器两个编译器，这里简要介绍集成环境编译器。所谓集成，是指编辑、编译、连接、运行集成在一个环境下。

1. 启动 Turbo C

只需执行 TC.EXE 文件，即可启动 Turbo C，进入 Turbo C 的操作界面。

Turbo C 是 DOS 程序，一般运行在 Windows 下的 DOS 环境。用 TC 加源程序文件名可快速启动 Turbo C，立即进入编辑操作。

2. Turbo C 操作

启动 Turbo C 后，主操作界面如图 1-2 所示。出现此界面是 Turbo C 启动成功的标志。

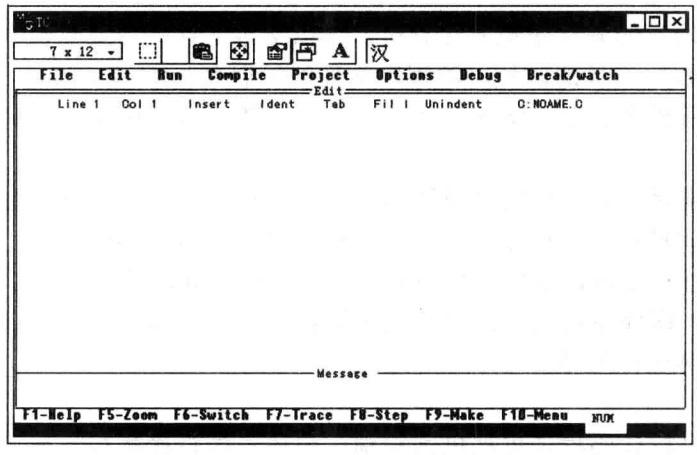

图 1-2 Turbo C 主操作界面

1）主操作界面

Turbo C 主操作界面由主菜单区、编辑窗口区、信息窗口区及热键区四部分组成。

Turbo C 的操作采用菜单驱动，主菜单包括 File（文件操作）、Edit（编辑操作）、Run（运行操作）、Compile（编译操作）、Project（工程文件）、Options（编译选择）、Debug（跟踪排错）等各项。

热键区提供常用操作的快捷功能键方式，如 F1（帮助）、F2（保存文件）、F3（载入文件）、F5（缩放窗口）、F6（切换窗口）、Alt+F5（用户屏幕）、Alt+X（退出 Turbo C）等。

2）主窗口

当前操作窗口为主窗口，亦称工作窗口。

主窗口为双线框标识，按 F6 可切换主窗口，按 F5 可缩放主窗口。

3）菜单选择

方式一：按 F10 进入主菜单区，通过菜单项的大写字母选择相应菜单项；或通过光标控制键移动高亮度光棒至相应菜单项，按回车键选择相应菜单项。

方式二：用 Alt 键加相应菜单项的大写字母，选择相应菜单项。

对子菜单项的选择，操作方法类似。

3．编辑

编辑在编辑窗口中进行，有多种方式进入编辑环境。

方式一：直接选择 Edit 菜单项。

方式二：选择 File 菜单项下的子项 New，用于建立新程序。

方式三：选择 File 菜单项下的子项 Load，载入要修改的程序，用于修改已存在的程序。

方式四：启动时采用快速启动方式。

Turbo C 的具体编辑操作是一般文字处理软件编辑操作的简单子集。实际上，源程序文件的编辑工作可以用任何文字处理软件完成。

编辑完成应检查是否有输入错误，发现错误及时改正。如无错，按 F2 或选择 File 项下的子项 Save 保存文件，选择 File 项下的子项 Write to 可将文件换名保存。

4．编译

Turbo C 将编译、连接一起完成。按 F9 或选择 Compile 菜单项进入编译。

编译可在内存中进行，亦可编译到磁盘。

编译时若发生错误，则需返回编辑环境修改程序，然后再编译程序。

5．运行

按 Ctrl+F9 或选择 Run 菜单项下的子项 Run，运行计算机中的程序。

按 Alt+F5 键或选择 Run 项下的子项 User Screen 进入用户屏幕，查看程序的运行结果。

运行时若发生错误，则需返回编辑环境，修改程序后，再编译、运行程序。

对于没编译的程序，选择运行会先自动进行编译操作，再运行程序。

6．退出

上机过程完成后，按快捷键 Alt+X 或选择 File 项下的子项 Quit 退出 Turbo C，返回操作系统状态。

退出 Turbo C 后还可用操作系统命令来显示源程序和运行程序。

其他上机操作环境参见《C 语言程序设计学习指导》。

习 题 一

1. 简述 C 语言的主要特点。
2. 简述标识符的命名方法，并与关键字进行比较。
3. 简述 C 语言程序的结构。
4. 简述 C 语言程序的实现步骤。

5. C语言程序的书写有何特点？
6. 注释在程序中有什么作用？
7. 下列标识符中，哪些是不正确的标识符，为什么？

   C SUM 3S Co.Ltd OK! com language for_c prg*1 do ￥123 (xyz) printf max PI China
8. 抄写例1-3中的程序，注意C语言程序的结构与书写格式。
9. 上机运行例1-1、例1-2、例1-3的程序，了解上机方法与步骤。
10. 模仿例1-1编写程序，输出自己的姓名、性别及年龄。
11. 模仿例1-2编写程序，求两个整数的积。
12. 模仿例1-3编写程序，求两个数的最小值。
13. 编写一个最小的C语言程序，并请上机运行。
14. 搜索、浏览C语言程序设计的相关网站。

第 2 章

基本数据类型

程序=算法+数据。

程序设计包括两个方面的工作，一方面对数据进行描述，另一方面对操作进行描述。数据是程序的加工对象，数据描述通过数据类型完成；操作描述通过语句完成，表达式完成操作的基本描述。

C 语言提供多种数据类型，基本类型有整型、浮点型、字符型、枚举类型和空类型五种，构造类型有数组类型、结构体类型、共用体类型和文件类型四种，此外还有处理地址数据的指针类型。其中，整型、浮点型、字符型、空类型由系统预先定义，又称标准类型，本章将全面介绍，并以此为基础介绍数据描述方法，即量的定义方法；其余为用户自定义类型，将从第 8 章开始全面讲述。

本章将从类型名称、取值范围、数据表示方法、基本运算操作、存储、标准函数及应用诸方面对数据类型加以介绍，对数据类型的掌握也应从这些方面着手。

2.1 整型

整型描述的数据是整数的一个子集。

2.1.1 基本整型

1. 类型名称

int。

2. 取值范围

–32 768～32 767，即-2^{15}～$2^{15}-1$。

3. 数据表示方法

与数学上类似，表示数据时不能有分隔符。

C 语言允许使用八进制整数与十六进制整数，八进制整数加前导符 0（数字），十六进制整数加前导符 0X（数字 0 和字母 X）。

4. 运算操作

整型数据能进行算术运算与关系运算。

（1）算术运算包括：+（加法）、–（减法）、*（乘法）、/（除法）、%（求余）。

运算规则除了除法外，与数学中的相同。