

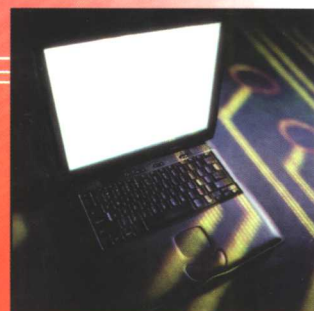
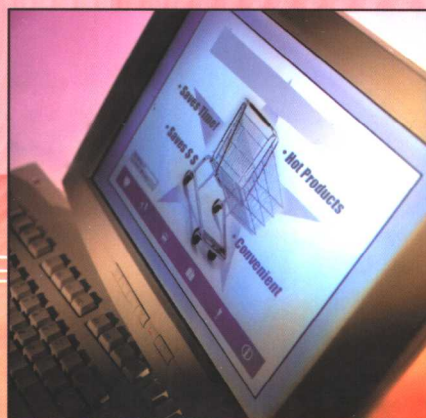
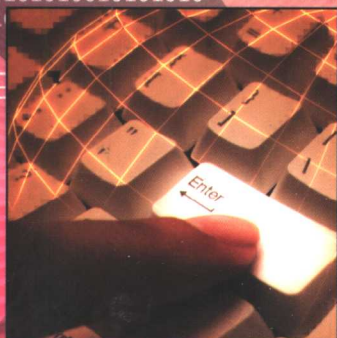
高等学校教学用书

C语言程序设计 教程

主编◎赵克林 姜春茂 韩忠东

0101101101010111010110100111011010101
1011011011011001111011011001011011010111010101
0100101110110101
10101101101001100101
101010
0101
01

10101010010101010
01



北京工业大学出版社

C 语言程序设计教程

主 编 赵克林 姜春茂 韩忠东
副主编 陈承欢 袁剑锋 吴建平

北京工业大学出版社

内 容 简 介

这是一本易学易懂的 C 教材。本书通过大量案例和习题,介绍了 C 语言的编程方法。主要内容有:初识 C 程序、C 语言基础、顺序结构、选择结构、循环结构、数组与结构、指针、函数与变量存储类别、文件、典型算法、综合实训等。

本书改变传统的 C 教材写法,突出了 C 语言的实用性、科普性和趣味性。本书可作为大中专、高职院校计算机专业或非计算机专业学生的学习教材,也是一本难得的培训教材和自学教材。

图书在版编目 (CIP) 数据

C 语言程序设计教程 / 赵克林, 姜春茂, 韩忠东主编. —北京: 北京工业大学出版社, 2004.8

ISBN 7-5639-1431-5

I .C… II .①赵…②姜…③韩… III .C 语言-程序设计-教材 IV .TP312

中国版本图书馆 CIP 数据核字 (2004) 第 089431 号

C 语言程序设计教程

主编 赵克林 姜春茂 韩忠东

*

北京工业大学出版社出版发行

邮编: 100022 电话: (010) 67392308

各地新华书店经销

徐水宏远印刷厂印刷

*

2004 年 8 月第 1 版 2004 年 8 月第 1 次印刷

787mm×1092mm 16 开本 16.25 印张 405 千字

印数: 0001~5000 册

ISBN 7-5639-1431-5/T·218

定价: 29.00 元

前 言

您是否为 Windows、Office 这样的软件拍手叫绝，认为大型游戏的设计者不可思议，或沉醉于因特网丰富 Web 页面中？这就是软件的魅力。当今世界，软件的发展方兴未艾。软件推动了硬件的发展，软件使我们的时代焕发了青春！软件人才是本世纪需要的四种紧缺型人才之一。我国的软件业较之美国、印度、日本尚有一定差距，这个领域大有作为，蕴藏着巨大的财富和就业空间。可以这么说：学语言前你的目的是用好别人的软件，学语言后，你则在思考怎样做出自己的软件了！

C 语言号称“黑客语言”，它面向过程，是所有计算机语言的基础，是迈入计算机软件殿堂的敲门砖，凡是有志于软件业者都应掌握它。但 C 语言又往往让初学者感到很难而望而却步，其实这与教材不无关系。本书突出了以下几方面的特色：

1. 与时俱进，注重知识的实用性。本书对 C 语言中那些过时的、偏僻的、对后继语言学习没有帮助的知识进行了大胆删除。

2. 重程序设计，而淡化语法。C 语言仅仅是一种表达工具，编写程序才是最终目的。

3. 将复杂知识简单化，抽象知识具体化。达到某一目的，我们不追求太多、太繁的方法，而只要求掌握一两种简单的方法即可。

4. 例题丰富、实用。C 语言到底能干什么？C 语言能解决我们生活中的现实问题！本书精选生活实例，既可提高读者兴趣，又能切实感到 C 语言之神奇，体现了“快乐学习”精神。

5. 深入浅出，逆向思维，教学互动，通俗易懂，是本书的编写特色。模仿编写程序→讲解涉及到的语法→阅读例程→参阅实例→完成作业，且对多数例题均提出了思考问题，这样，引人入胜、步步提高，让读者在“不知不觉”中学懂 C 语言。

6. 为了全面提升读者编程水平，本书的画龙点睛之处是第 10 章介绍了编程的部分典型算法，第 11 章接着用 C 语言开发了一个“五子棋”游戏软件，以此综合运用全书知识。

本书由赵克林、姜春茂、韩忠东担任主编，由陈承欢、袁剑锋、吴建平担任副主编。参加本书编写的有（以姓氏笔画为序）：马伟强、云贵全、许大荣、刘长春、刘建国、许沪敏、李武、杨俭、张邦文、吴建平、陈承欢、贺涛、赵克林、姜春茂、袁剑锋、韩忠东、裴来芝。全书由赵克林、陈承欢负责组织、统稿和审核。

由于作者水平有限，书中疏漏错误之处在所难免，敬请读者指正。

编 者

2004 年 8 月

目 录

第1章 初识 C 程序	(1)
1.1 了解 C 程序	(1)
1.2 怎样才能学好 C 程序	(3)
1.3 C 集成上机调试环境简介	(6)
第2章 C 语言基础	(9)
2.1 C 语言数据结构	(9)
2.2 常量与变量	(10)
2.3 运算符	(15)
2.4 常用数学函数及使用技巧	(22)
2.5 表达式	(25)
2.6 小结与实例	(27)
第3章 顺序结构程序设计	(33)
3.1 顺序结构程序流程图	(33)
3.2 直接赋值语句	(35)
3.3 键盘输入语句	(36)
3.4 输出语句	(37)
3.5 小结与实例	(40)
第4章 选择结构程序设计	(44)
4.1 模仿编写 C 选择程序	(44)
4.2 if 语句	(48)
4.3 switch 语句	(54)
4.4 小结与实例	(63)
第5章 循环结构程序设计	(66)
5.1 if 与 goto 构成循环	(66)
5.2 for 循环	(70)
5.3 while 循环	(85)
5.4 do - while 循环	(90)
5.5 小结与实例	(96)
第6章 数组与结构	(115)
6.1 一维数组	(115)
6.2 字符数组	(125)
6.3 结构	(133)
6.4 二维数组	(139)
6.5 小结与实例	(145)

第 7 章 指针	(155)
7.1 指针的概念	(155)
7.2 指针类型	(156)
7.3 指针与数组	(162)
7.4 小结与实例	(171)
第 8 章 函数与变量存储类别	(178)
8.1 模仿编写自定义函数程序	(178)
8.2 自定义函数	(181)
8.3 变量作用域与存储类别	(190)
8.4 数组、指针作函数参数	(192)
8.5 小结与实例	(202)
第 9 章 文件	(211)
9.1 C 文件概述	(211)
9.2 文件操作语句	(212)
9.3 小结与实例	(218)
第 10 章 部分典型算法	(224)
10.1 算法的概述	(224)
10.2 典型算法简介	(227)
第 11 章 C 程序综合实训（五子棋游戏）	(234)
11.1 实训目的	(234)
11.2 程序编写步骤	(234)
11.3 程序功能分析	(234)
11.4 定义核心数据结构	(234)
11.5 程序的模块划分	(235)
11.6 五子棋游戏操作说明	(235)
11.7 实训内容与学时分配	(236)
11.8 教学组织设计	(236)
11.9 考核方式与标准	(236)
11.10 参考源程序	(237)
11.11 源程序中用到的标准函数说明	(248)
附录 1 ASCII 表	(250)
附录 2 C 保留字	(251)
附录 3 C 运算符	(251)
附录 4 C 程序调试错误信息	(251)

第1章 初识 C 程序

1.1 了解 C 程序

先看两个实例。

【例 1-1】 已知正方形边长为 5，求周长、面积。

请想想：数学上怎么做？若设边长为 a 、周长为 l 、面积为 s ，则 $l = 4a$ ， $s = a^2$ 。

我们写 C 程序让计算机帮忙：

```
main()
{int a,l,s;    /* 定义 a, l, s 三个整型变量 */
  a=5;         /* 边长为 5 */
  l=4*a;       /* 算周长 */
  s=a*a;       /* 算面积 */
  printf("\nl = %d,s = %d",l,s);    /* 输出结果 l, s */
}
```

进入 C 语言系统，录入这段程序（/*...*/之间的内容表示注释，可省略），执行（Ctrl + F9），输出结果为：

$l = 20, s = 25$

不管以前是否学过 C 语言，模仿本例，你应该能写出已知正方体的边长为 10，求表面积和体积的程序：

```
main()
{int a,s,v;    /* 定义 a, l, s 三个整型变量 */
  a=5;         /* 边长为 5 */
  s=6*a*a;     /* 算表面积 */
  v=a*a*a;     /* 算体积 */
  printf("\n%d,%d",s,v);    /* 输出结果 l, s */
}
```

请观察：这两个程序与数学有什么联系吗？

- (1) 公式与数学上完全相同。
- (2) 编程思路与计算机工作原理一致（输入数据、处理数据、输出结果三步）。
- (3) 用分号隔开的称为语句，为解决某一问题的若干语句的有机组合便构成了程序。
- (4) 运算符换了写法。我们注意到乘法运算符“ $\times$ ”换成了星号“ $*$ ”。为什么？如果

乘号仍用“×”表示，容易与字母 x 混同。类似地“÷”换成“/”，也就是说运算符要作必要的变化。

(5) 多了几个单词。

main(): 称为主函数，C 语言号称函数语言，在众多函数中，main() 是最特殊的，无论程序多长，主函数只能有一个，程序执行时开始于主函数也结束于主函数，换句话说，其他函数都是被 main() 调用的。

int: integer 的缩写，整数的意思。它告诉系统，后边的几个变量是整型的，小数将被忽略。有人对“C 程序使用变量前必须先定义其类型”感到很烦，但这正是 C 程序的优点！为什么 C 程序那么节约系统资源、程序执行效率那么高、可移植性那么强呢？与此大有关系。不同类型的变量占用内存空间是不一样多的，C 程序刚好能做到物尽其用，适可而止。当然，如果变量是实数类型，int 就要换成 float 了。

printf(): 这是一个输出函数，负责把程序运算的结果输出到屏幕，供人观看。%d 表示对应的值取整数。如果是实数，则用%f表示。

C 语言一共只有 32 个这样的单词（保留字），只要掌握了 10 来个，便可以随心所欲地编写程序了！

还有一对花括号 {}，表示若干语句组成的整体。建议左“{”和右“}”纵向对齐，以方便查错。另外，“}”后没有分号。

(6) 行后有注解。对行内容适当注解，以方便阅读。注释内容放于“/*”和“*/”之间，是不执行的，可有可无。一般，对关键程序行才要求注解。

C 程序一行可写多个语句，字母要区分大小写。作为良好的编程习惯，程序外观最好成锯齿形，变量名见名知义或符合数学学习惯，算法精益求精、追求个性，关键语句尽量注释。总之，程序如人，一道程序是一个人综合素质的体现。

【例 1-2】 输入圆的半径，求它的面积。试写 C 程序。

程序需要两个变量：半径 r 和面积 s ，应该是实数类型。程序如下：

```
main()
{
    float r,s;           /* 定义 r, s 两个实型变量 */
    scanf("%f",&r);      /* 键盘输入半径 r */
    s = 3.14*r*r;        /* 算面积 */
    printf("\n%f",s);    /* 输出结果 s */
}
```

程序执行时，从键盘上输入 r 值，立即输出对应的圆的面积，显然较【例 1-1】通用些。

请模仿前例，写出下面两个程序：

(1) 设两个电阻 R_1 、 R_2 的阻值分别为 $100\ \Omega$ 、 $200\ \Omega$ ，求它们并联后的电阻。试写 C 程序。

(2) 已知爸爸质量 $100\ \text{kg}$ ，儿子质量 $15\ \text{kg}$ ，相距 $20\ \text{m}$ ，求父子俩之间的万有引力。已知万有引力常数 $G = 6.67 \times 10^{-11}$ 。试写 C 程序。

UNIX 网络系统（第 5 版）便是由它开发的。下边二例是输出两张图形。

【例 1-3】 打印楼梯，同时在楼梯上方打印两个笑脸。

```
#include "stdio.h"
```



```

main()
{int i,j;
 printf("\1\1\n"); /* 输出两个笑脸 */
 for(i=1;i<11;i++)
 {
     for(j=1;j<=i;j++)
         printf("%c%c",219,219);
     printf("\n");
 }
}

```

【例1-4】 显示一张国际象棋的棋盘。

```

#include "stdio.h"
main()
{int i,j;
 for(i=0;i<8;i++)
 {
     for(j=0;j<8;j++)
         if((i+j)%2==0)
             printf("%c%c",219,219);
         else
             printf(" ");
     printf("\n");
 }
}

```

C程序还可方便地实现对文件的管理，下例将在磁盘上新建文件夹。

【例1-5】 程序运行时，在当前文件夹下建立子文件夹 score，在C盘根文件夹下建立文件夹 student。

```

#include "dir.h"
main()
{mkdir("score");
 mkdir("c:\student");
}

```

1.2 怎样才能学好C程序

C语言对于初学者，是迈入软件之门的第一道通行证。C语言简洁紧凑、小巧灵活、书写自由，32个保留字、若干个库函数以及字母数字构成了它的全部符号集；它是一种中级语言，既有高级语言的全部特征，又能直接对硬件操作；它是面向过程语言的典型代表，支持由顶向下、分解任务的现代结构化程序设计，方便多个程序员合作开发软件；它具有丰富的数据类型、运算符和各类库函数，极大地缩短了程序长度，便于学习。另外，C程序还有

易移植的特点，基本上可在所有计算机上运行。

怎样才能学好 C 语言呢？

学 C 语言很容易进入两个误区：

- ① 纯粹学语法。
- ② 只看不写。

譬如学英文，一个只懂语法而不会说英文的人算不算学好了英文呢？答案是否定的。计算机语言与人类语言一样，种类繁多，但都是表述一个事件的工具。

学语言的目的是为了写程序，学语法或看别人的程序是为了帮助我们更好地写程序。程序是写会的，不是看会的。

但这并不是说，语言人人都可学会。好的教材、好的学习方法及充分的上机时间会使学习语言事半功倍。写程序并不神秘，它只不过是把我们平时解题的步骤用计算机的规则描述出来而已。从宏观上说，结构化程序设计的步骤如图 1-1 所示。

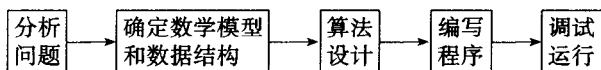


图 1-1 结构化程序设计步骤

① 分析问题：分析题意，搞清楚已知什么、求什么。

② 确定数学模型和数据结构：把实际问题转化成数学问题，并确定使用到的各个变量的类型。

③ 算法设计：算法是程序的灵魂，解一个题有哪些方法，哪种方法是最好的？这要从程序占用内存空间、程序执行时间、程序的可读性和可移植性等几方面综合考虑而定。

④ 编写程序：选择一种计算机语言，按照算法写出源程序。

⑤ 调试运行：利用计算机调试程序，纠正错误，编译连接生成目标代码（或可执行文件）。对初学者，最好先手工写好源程序，然后再上机调试；但当熟练到一定程度后，则适宜边写边调，一步完成。

1.2.1 尽量画流程图

流程图分为传统流程图和 NS 结构流程图两种，它是算法的描述工具。传统流程图直观明了，好学好用，本书的流程图一律采用传统流程图。NS 流程图请读者参考有关书籍。

传统流程图符号有四框一线，如图 1-2 所示。



图 1-2 传统流程图符号

图 1-3 是【例 1-2】的流程图。

流程图描述的是解题步骤或某种算法，框内可用汉字、数学表达式、C 语句等没有限制。按图索骥，将每个框转换为一个或多个语句便成了程序。所以，程序 100 分，流程图占 90 分，语言占 10 分。比如图 1-3 所示的流程图，QB、C++、DelPhi 随便选什么计算机语言都可写出程序。

结构化程序设计把程序结构分为三种：顺序结构、选择结构、循环结构，这三种程序结构都可方便地用流程图描述。图1-3为顺序结构。

1.2.2 追求最佳算法

什么是算法？算法就是解题的方法。一个典型例子是求：

$$s = 1 + 2 + 3 + \cdots + 100$$

最笨的方法（也是最易想到的方法）是先算出 $1 + 2 = 3$ ，再算 $3 + 3 = 6$ ，再算 $6 + 4$ ，依次往后加，直到加到100。可能要花20分钟，结果还不一定准确。

最好的方法是利用交换律，改写表达式为： $s = (1 + 99) + (2 + 98) + \cdots + (49 + 51) + 50 + 100 = 49 \times 100 + 50 + 100 = 5050$ ，只需花20秒，且结果准确。

因此，最好的算法往往不是最容易想到的算法。以后我们写程序不要看题即写，而要先思考算法，从若干算法中选一种最好的算法，画出其流程图，再照流程图写程序。

1.2.3 良好的编程风格

程序员有编程的风格，特别是大型或较大型程序，多人合写，程序段互相交流，良好的编程风格尤其重要。

下面是本书程序将遵守的风格：

- ① 合理加入空行。各自定义函数之间、功能相对独立的程序段之间宜加一空行相隔。
- ② 适当加入空格。关键字之后、二元运算符的前后宜加一个空格。
- ③ 同类变量的定义、每一条语句各占一行，便于识别和加入注释。
- ④ 变量赋初值采用就近原则，最好在定义变量的同时赋以初值。
- ⑤ 选择结构的 if、else、switch，循环结构的 for、while、do 等关键字加上其后的条件、括号独占一行，并且“{”或“}”独占一行或合占一行，以保持括号配对。
- ⑥ 多层嵌套结构，各层应缩进对齐（2格），且每层的“{ }”应严格垂直左对齐，以保持嵌套结构的层次关系一目了然，便于理解。
- ⑦ else 应与其配对的 if 对齐，以免引起误解。
- ⑧ 语句不宜太长，不要超出人的视力控制范围。如果语句太长，应断行，但须在上行尾使用续行符“\”。
- ⑨ 标识符的命名要么符合数学习惯，要么见名知义（或英文或拼音）。符号常量全用大写字母。指针变量名加前缀“p”，文件指针变量名加前缀“fp”。
- ⑩ 每一个程序都按“函数原形→预处理→主函数→自定义函数”的顺序编写。

1.2.4 语法娴熟，多解习题

语法虽然不是第一位的，但不懂语法也写不出程序。学好计算机语言的惟一途径是多解习题，并且一题多解。

1.2.5 注重上机实践

只有计算机能够执行并且结果合理的程序才是正确的程序，这就要上机调试。对C程

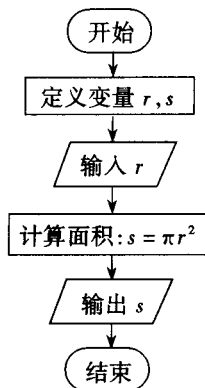


图1-3 例1-2的流程图

序来说, 程序执行时有输出, 顶多说明程序没有语法错误, 编译通过了。结果是否符合实际, 只有我们人为检验。

1.3 C 集成上机调试环境简介

1.3.1 C 环境介绍

C 语言系统的文件目录如下:

C:\TC\TC.EXE (主文件)

| - INCLUDE

| - LIB

| - USER

C 语言集成上机调试环境非常好用, 只需掌握四个快捷键就可调试 C 程序了。

① F10: 激活主菜单 (或 Alt + 主菜单的高亮字母, 如 Alt + E)。

② Ctrl + F9: 编译执行源程序。

③ Alt + F5: 浏览屏幕输出结果。

④ Alt + X: 退出 TC 2.0。

Turbo C 2.0 语言集成调试环境界面如图 1-4 所示。界面主要由主菜单、源程序编辑区、信息窗口 Message 组成。信息窗口用于显示出错信息, 建议保留 (可在主菜单 “Options” 中关闭)。

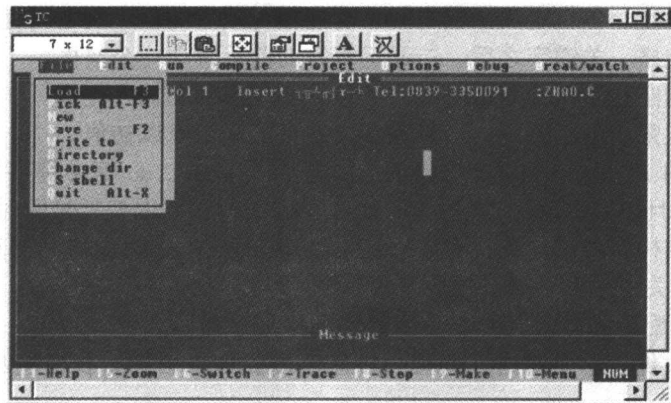


图 1-4 Turbo C 2.0 “File” 菜单

作为进一步掌握编程环境, 还需了解 “File” 菜单和 “Option” 菜单 (见图 1-5)。

“File” 菜单下子菜单主要了解: “Load” (调入源程序)、“New” (清除源程序)、“Save” (存盘)、“Write to” (改名存盘)、“Change dir” (设用户文件存放目录)、“Os shell” (回到 DOS 环境, 输入 exit 返回 TC 2.0, 重要!)。

“Options” 菜单下子菜单主要了解: “Directories” (分别设置 C 语言头文件、库函数、可执行文件存放目录。很重要!)、“Environment” (C 语言窗口是否包括信息窗口 Message)。

另外, C 语言是一种编译型计算机语言, 源程序的扩展名为 .c, 编译后目标代码文件扩展名是 .obj, 生成的可执行文件扩展名是 .exe。如图 1-5 所示, 源程序 zhao.c, 则生成的目

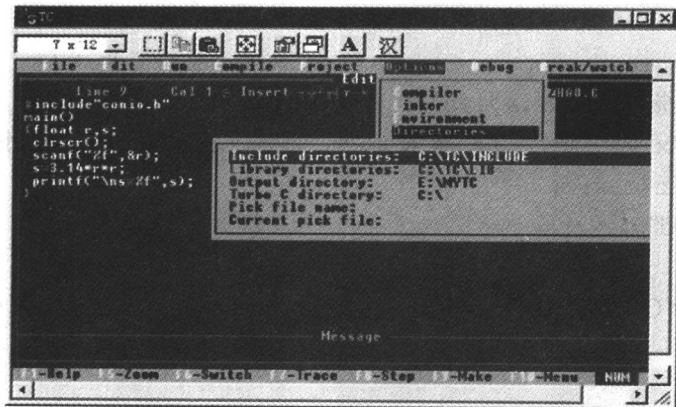


图 1-5 Turbo C 2.0 “Options” 菜单

标文件名为 zhao.obj，可执行文件名为 zhao.exe。

1.3.2 C 程序调试步骤

如图 1-5 源程序 zhao.c，其功能是输入圆的半径求面积。调试步骤如下。

第一步：进入 C 语言，设置环境。主文件名 TC.EXE，从 DOS 或 Windows 环境都可进入。环境设置主要是指“Options”菜单下“Directories”前两项：第一项“Include directories”设头文件（文件包括）路径；第二项“Library directories”设库函数路径。无论 C 语言装在哪个磁盘（C 盘、D 盘等），这两个路径一定要与实际情况相符。

第二步：输入源程序。

第三步：修改调试。结合“Ctrl + F9”、“Alt + F5”两键边执行边查看，同时根据信息窗口（Message）提示边修改，如此反复，直到成功。如有必要，最后还需保存源程序。

为了保证 C 程序的正常调试，在此向读者建议几点：

- ① 每次调试的源程序要取不同的文件名。
- ② 调试程序前一定要清掉以前的程序（即不能在一个源程序后接着输入第二个源程序）。
- ③ 注意字符的大小写。
- ④ 十分留心信息框内的错误提示，提高调试速度。常见错误提示见附录 4。
- ⑤ 如果是汉字环境，除了汉字外其他任何字符只能在半角状态输入。
- ⑥ 源程序、目标文件、可执行文件放于指定的个人目录中。

习 题 1

1. 在 C 语言中，源程序文件的后缀是_____，经过编译后的文件后缀是_____，经过连接后的文件后缀是_____。
2. 结构化程序是由_____、_____、_____三种结构组成。
3. 在一个 C 语言的源程序中，必不可少的一个函数名是_____。
4. 请画出下面程序的流程图：

```
main()
{ int x,y,z,max;           /* 定义 4 个整型变量 */
  scanf("%d%d%d",&x,&y,&z); /* 通过键盘输入实现对 x、y、z 赋值 */
  max = x > y ? x : y;      /* 如果 x > y, 则把 x 值赋给 max, 否则赋 y 值给 max */
  max = max > z ? max : z;
  printf("%d",max);        /* 输出 max 的值 */
}
```

5. 编译并执行 C 源程序使用快捷键_____, 查看屏幕输出结果使用快捷键_____。

第2章 C语言基础

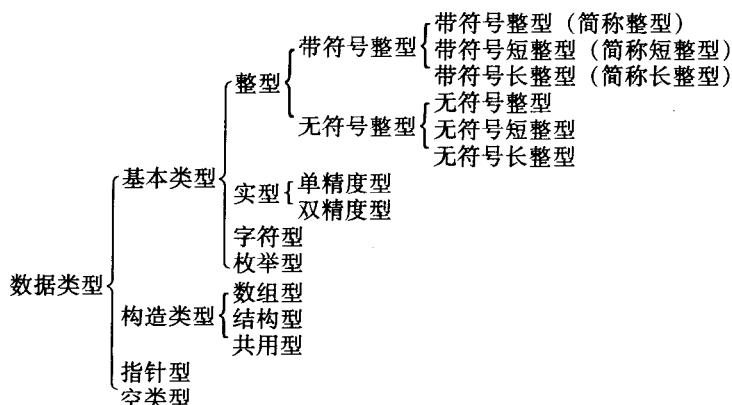
任何语言都有一定的规则。一般使用高级语言编写程序，不外乎在程序中做两件事：一是描述数据，二是描述作用在这些数据上的操作。前者通过数据定义语句实现，后者通过若干执行语句实现。C语言也不例外。本章主要介绍C语言数据类型、常量、变量、运算符、函数、表达式的规则，这些是后续各章节的基础。至于复杂的数据类型如数组、指针、结构型等，将在以后的章节中陆续介绍。

2.1 C语言数据结构

在商店中出售的衣服有S、M、L、XL、XXL等几种尺码可以满足不同人的身材大小。在计算机中存储、处理数据，也要将数据分成几种类型，就像对于不同身材的客户，服装工厂要提供不同的模板，生产不同的衣服号码一样。C语言程序中使用的数据都属于特定的类型，没有类型的数据是不存在的。例如，职工的年龄和工资都可以进行加、减等算术运算，具有一般数值的特点，在C语言中称为数值型。其中年龄是整数，所以称为整型；工资一般为实数，所以称为实型。又如职工的姓名和性别，是不能进行加、减等算术运算的，这种数据具有文字特征，姓名是由多个字符组成的，在C语言中称为字符串；性别可以用单个字符表示（例如用M表示男性，F表示女性），在C语言中称为字符型数据。如果某数据在程序运行中，只取固定的几个值，我们就可以把这几个值列出来，以后这个数据只能取这几个值中的某一个。这种数据在C语言中就叫做枚举型。

我们把整型和实型合称为“数值型”，把数值型、字符型、枚举型合称为“基本数据类型”。C语言根据数据加工处理的特征，还设有其他复杂的数据类型，具体如表2-1所示。

表 2-1 C语言的数据类型



构造类型是指由若干个相关的基本类型数据组合在一起形成的一种复杂的数据类型。数组是由相同类型数据组合而成的，例如，若干个人的年龄组合在一起，就是一个整型数组；若干人的基本工资、职务工资、奖金组合在一起，就是实型数组。结构型是由不同数据类型组合而成的，例如，一个人的姓名（字符串）、性别（字符型）、年龄（整型）、基本工资（实型）也可以组合在一起，构成一个结构型数据。如果若干个数据不同时使用，我们就可以让它们占用相同的内存区域，以便节省内存，这些数据组合在一起就是“共用型”。由于计算机硬件技术的飞速发展，内存越来越大，共用数据类型已没有多少实际意义，故本书不讲共用数据类型。

指针型是一种简单的数据类型，它是用来表示内存地址的。指针类型的数据可以表示基本类型数据的地址，也可以表示结构类型数据的首地址和其中某个具体数据的地址。例如，存放一个人年龄的内存的地址、存放某数组的首地址等都可以用指针型数据来表示。

空类型是从语法完整性的角度给出的一种数据类型，表示该处不需要具体的数据值，因而没有数据类型。

对每个数据，C 语言系统都要在内存中（个别数据可能在寄存器中）为其分配若干字节，用于存放它们。数据所占用的内存字节数称为该数据的“数据长度”。不同类型数据的长度是不同的，因此 C 语言规定每使用一个数据之前，必须对数据的类型加以定义，以便为其安排长度合适的内存。

2.2 常量与变量

2.2.1 常量

常量是 C 语言常用的操作数，如 2、3.5 等，它们在程序运行过程中保持数值不变。常量可分为一般常量和符号常量两类。

1. 一般常量

这是应用最多的一类常量，有整型、单精度型、双精度型、字符型（串型）等类型，C 系统根据其常量外观形式自动识别。如 3.5 为单精度型，76 为整型，'A' 为字符型，"computer" 为字符串型。

对整型常量，又有十进制、八进制、十六进制之分。系统默认十进制，如 327；八进制常数以数字 0 开头，如 0756；十六进制常数以 0x 开头（数字 0 和小写字母 x），如 0x56a。整型常量范围是 $[-32768, 32767]$ 。

实数在 C 语言中有两种写法：十进制形式、指数形式。指数形式适宜表示较大或较小的数，如：

-1.6×10^{-19} ：可表示为 $-1.6e-19$ 、 $-1.6E-19$ 、 $-16e-20$ 等。

6.02×10^{23} ：可表示为 $6.02e23$ 、 $0.0602e25$ 等。

其中，e 或 E 代表底数 10，它的前边可以是任意小数，后边只能是整数。如：E2、1e、1e2.5 等属不合法的 C 常数。

字符型常量，有两种写法，如 'A'、"hellow"。单引号引起的只能是一个字符，双引号引起的可以是多个字符，称字符串常量。这里，单引号、双引号分别只作为字符或字符串的定界符，并不是字符常量、字符串常量本身的内容！

特殊地,除了显式字符常量外,还有一些专用字符或者屏幕上无法显示的字符(这些无法显示的字符代表着某些操作),如'\a'、'\n',分别产生警告声和换行,这类特殊的常量字符在C语言中称转义字符,详见第3章。

【例2-1】 下列C常量中哪些是合法常量?

0.567、010、095、0x10、0.1e0.1、1/2

【分析】

095 不合法,因为0开头代表八进制数,而八进制中最大数字是7。

0.1e0.1 不合法,e后指数不能为小数。

1/2 不全法,所谓常量指不能再运算的量,1/2含运算符,是表达式不是常量。

2. 符号常量

定义形式: #define 常量名 常量值

如 #define PRICE 35.5

符号常量指把常量赋给一个符号,这个符号可理解为常量的别名。符号常量一旦定义,在整个程序内便不能再改变其值。在程序中适当设置符号常量,不但使程序简洁,更可避免一错全改的麻烦,非常有助于我们快速调试程序!

注意:定义符号常量行后无分号;一般符号常量用大写字母表示。

例如:当半径为2.0时,要计算圆的面积和周长,可以使用语句:

```
s = 3.1415926*2.0*2.0;
```

```
l = 2*3.1415926*2.0;
```

现在要把半径改为5.0,而程序中有3个2.0,要修改哪个呢?漏改一处就出错。这时我们如果使用了符号常量,当半径再次修改时,只需改变其定义行即可,显然简单多了,见例2-2。

【例2-2】 符号常量在程序中的应用。

```
#define Pi 3.1415926
#define R 5.0
main( )
{float s,l;
  s = Pi*R*R;
  l = 2*Pi*R;
  printf("s = %f,l = %f",s,l);
}
```

另外,C语言还提供一种“普通常量”的方法进行类似处理,请看下面的例子并与上例比较。

【例2-3】 普通常量在程序中的应用。

```
#include "conio.h"
main( )
{float s,l;
  float r = 5.0;
  const float Pi = 3.1415926;      /*常量 Pi 表示 3.1415926 */
  clrscr();
  s = Pi*r*r;
```