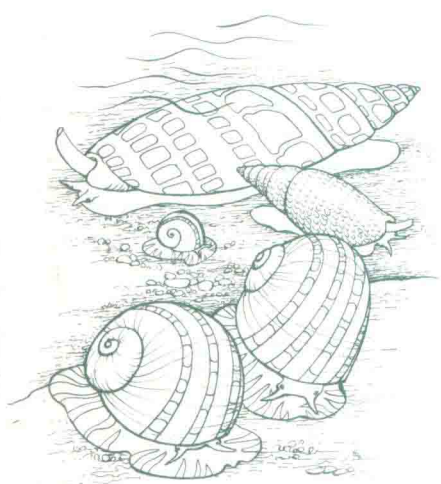
工业和信息化普通高等教育“十三五”规划教材立项项目
21世纪高等学校计算机规划教材
21st Century University Planned Textbooks of Computer Science

C语言程序设计与应用实验指导书（第2版）

Experimental Guide of Programming and Application of the C Language (2nd Edition)

张小东 主编
郑宏珍 主审

- 全面指导C语言程序设计与编程
- 注重培养实际设计与编程的能力
- 满足高级程序语言设计教学需要



高校系列

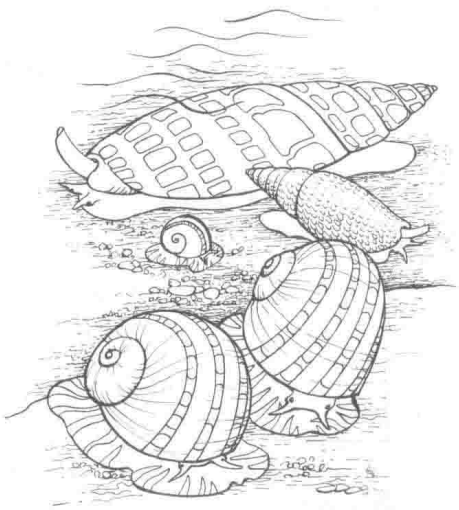
中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

工业和信息化普通高等教育“十三五”规划教材立项项目
21世纪高等学校计算机规划教材
21st Century University Planned Textbooks of Computer Science

C语言程序设计与应用实验指导书（第2版）

Experimental Guide of Programming and Application of the C Language (2nd Edition)

张小东 主编
郑宏珍 主审



高校系列

人民邮电出版社
北京

图书在版编目（CIP）数据

C语言程序设计与应用实验指导书 / 张小东主编. -- 2版. -- 北京 : 人民邮电出版社, 2017.9
21世纪高等学校计算机规划教材
ISBN 978-7-115-46920-5

Ⅰ. ①C… Ⅱ. ①张… Ⅲ. ①C语言－程序设计－高等学校－教学参考资料 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2017)第228959号

内 容 提 要

本书与人民邮电出版社出版的《C 语言程序设计与应用（第 2 版）》一书相配套，主要内容包括：各章学习辅导与习题解答、实验指导与实验报告。各章学习辅导与习题解答部分共包含 9 章、2 套模拟试题及解答，其中每章又分为本章学习辅导、课后习题指导以及实验问题解答 3 部分。实验指导与实验报告部分共包含 8 个实验，每个实验又分为实验目的、实验指导、实验内容和实验小结 4 部分。

本书将学习辅导与实验指导相结合，内容丰富、重点突出、设计新颖、讲解详尽，能帮助初学者快速且扎实地掌握 C 语言知识，不仅可作为高等院校 C 语言课程的配套教材，还可作为广大计算机技术人员及相关自学者的辅助教材。

◆ 主　　编　张小东
主　　审　郑宏珍
责任编辑　张　斌
责任印制　陈　犇

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：7.75　　2017 年 9 月第 2 版
字数：326 千字　　2017 年 9 月北京第 1 次印刷

定价：36.00 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

本书编审人员

主　审　郑宏珍

主　编　张小东

副主编　张维刚　张　华　李春山　周学权

参　编　向　曦　马　帅　刘艺姝　张壹帆　崔　杨　倪　烨

　　过友辉　衣景龙　张天昊　张博凯　杨　帆　刘　娟

前　言

C 语言结构化、简单、灵活、可移植等多个优良特点，决定了其在程序设计中的基础性地位，在教学中有难以动摇的实际应用。作为大多数学生第一种需要认真学习理解的编程语言，C 语言已成为他们中间很多人的“编程母语”，深深地烙印在学生的思维方式中。其中，实验是学习 C 语言最为重要的一个环节，学生通过实验把课堂上学到的理论知识运用于实践当中，建立对程序的基本认识和对计算机模型的最初理解。为了帮助读者尽快掌握 C 语言的初步编程方法和程序设计思维，我们特地编写了这本与《C 语言程序设计与应用（第 2 版）》配套的实验指导书，以便同学们在完成一定量的课程及课外项目实践后，建立正确的软件开发实践习惯。本书共分两大部分：第一部分为学习辅导与习题解答，第二部分为实验指导与实验报告。

在第一部分学习辅导与习题解答中，按照《C 语言程序设计与应用（第 2 版）》中的各章进行学习辅导，每章分为 3 个模块，第一个模块为本章学习辅导，对本章所涵盖的知识点进行了汇总，按词汇、语法和应用的线路进行辅导，并对关键的内容、编程技巧和易错、易混、易乱的知识点进行了要点提示；第二个模块为课后习题指导，包括每一章课后的习题正确答案和较为详细的解释，特别是对编程题，一般是以问题分析、算法设计和代码实现等软件算法设计方法学的基本思想为指导进行解答；第三个模块为实验问题解答，融合了多位在 C 语言教学一线工作的教师多年在指导学生实验方面的经验，总结学生在实验过程所遇到的典型问题，做了较为详尽的解答，以帮助读者更好地进行实验。另外，在这部分还安排了两套精心编制的试卷和详尽的试题解答，使同学们能够对自己的学习情况进行检查。

在第二部分实验指导与实验报告中，从教材的第 2 章开始设置了 8 个实验，每个实验分 4 个模块，第一个模块实验目的，是完成本次实验后所要达到的目标，即了解什么，熟悉什么，掌握什么；第二个模块实验指导，说明了完成本实验所需要的参考学时数（每学时为 50 分钟），针对本次实验中所遇到的难点和编程技巧进行辅导；第三个模块实验内容，按照每章所涉及的知识点精心编制实验题目，其中包括阅读程序、编程并上机调试、调试记录，旨在帮助同学们运用教材上学到的知识进行实践演练，尽快掌握本章的知识点，同时养成良好的编程习惯；第四个模块实验小结，以自检表的形式将本章所涉及的知识点用问题展示出来，读者按照自己的实际学习情况如实回答，每张自检表有 3 次回答机会，对于第一次没有掌握好的，经过复习准备后，再进行第二次、第三次回答，以确定对每章知识点的掌握程度。

本书的主要特点有以下几点。

1．避免机械思维，变被动学习为主动学习

对于刚刚接触 C 语言学习的学生来说，开始实验时相对比较慌乱，往往是机械而盲目地将指导书上的代码敲入计算机中，验证代码的正确性，而忽视了实验最重要的目的是学习如何运用 C 语言去设计程序，并非代码验证！这种被动的学习方式通常很难达到实验所期待的教学效果。因此，在阅读程序题这一模块中加入程序扩展和扩展分析等内容，旨在帮助同学们从机械的思维中解脱出来，主动思考在程序扩展的变化中本段代码的含义，学习如何进行代码的分析与设计，在潜移默化中变被动学习为主动学习。

2．加强实验中的互动性，提高独立解决问题的能力

实验问题解答模块针对实验所涉及的题目和实验中同学们容易出现的错误，列出了诸多问题，并进行了详细的解答，尽最大努力帮助同学们做好实验。它采用了实验→问题→思考→解答→实践的良性循环思维模式，体现了本书的互动性，在提高同学们独立解决问题能力的同时，也减轻了指导教师的工作负担。

3．突出程序设计思路和程序设计表达方面的培养

学习语言的最终目标是能够进行正确的程序设计并能表达出程序设计的思想，以便进行交流、改进和维护。所以，本书从一开始便注重对学生正确程序设计思维的培养和训练，每道编程题都是以问题分析→流程图的绘制→代码编写→测试与分析的流程模式进行讲解，同时在实验内容的设计中也要求学生按照这一线路进行训练，以达到预期的目标。

4．抓住学习重点，提高自学能力

为了让学生抓住学习重点，提高学习效率，本书除了设置了学习辅导、习题指导、实验问题解答和丰富的实验内容及指导外，还有一个实验小结模块，汇总各章节的知识点内容，以问题的方式提出，帮助学生理清学习思路，把握学习方向，提高自学能力。

5．进行初步工程能力方面的培养

本书在程序设计时，按正向工程模式训练，即按照问题分析→模型建立→算法描述（流程图）→算法实现（程序）→测试→编写使用手册的流程进行；在阅读程序时，按照反向工程模式培养，即通过进行程序扩展与结果分析，推导程序解题的设计方法和数学模型。这两种能力都是系统分析师、设计师或程序员所必须具备的能力，需要进行必要的训练与培养。

全书由张小东负责统稿，第 1、2、4 章由张小东编写，第 3、6 章由张维刚编写，第 7、8 章由张华编写，第 5 章由李春山编写，第 9 章由周学权编写。郑宏珍教授在百忙之中审阅了全部初稿，对本书提出了很多宝贵意见。在书稿的录入、校对及实验内容、例题和习题的审核调试过程中，向曦、马帅、刘艺姝、张壹帆、崔杨、倪烨、过友辉、衣景龙、张天昊、张博凯、杨帆、刘娟等同志也做了大量的工作。

因编者水平有限，书中疏漏在所难免，恳请读者批评指正。作者 E-mail 为 z_xiaodong7134@163.com，wgzhang@jdl.ac.cn。欢迎读者给我们发送电子邮件，对本书提出宝贵意见。

编　者

2017 年 8 月

目 录

第一部分 各章学习辅导与习题解答

第一部分
各章学习辅导与习题解答

第1章 简单C程序设计

1.1 本章学习辅导

1.1.1 C语言程序的结构

C语言程序的结构共分4部分：注释、预处理指令、main函数、其他自定义的函数及语句。

（1）注释：包含在符号“/*”和“*/”之间（可有多行）或跟在“//”之后无换行的文字。它是进行功能说明的非C语言语句，是不会被执行的部分。

（2）预处理指令：本章只介绍#include 指令，它将包含在当前目录或系统目录下的头文件引入本文件中。#include 后面跟< a.h >表示在包含系统头文件的目录（通常就是C语言程序的安装路径）下找此头文件a.h，#include 后面跟“”表示先在当前目录下找此头文件，若找不到，再到系统目录下找。

（3）main函数：C语言程序起始于main函数的“{”，结束于main函数的“}”；每一个C语言程序有且只能有一个main函数。

（4）其他自定义的函数及语句：由程序员按C语言的语法规则自己定义的函数或语句。

1.1.2 C语言中的符号规定

（1）关键字：又称保留字，它是C语言中预先规定的、具有固定含义的一些单词。

（2）标识符：指常量、变量、语句标号以及用户自定义函数的名称。使用时，要注意以下几点。

① 所有标识符必须由字母（a～z，A～Z）或下划线（_）开头。

② 标识符的其他部分可以由字母、下划线或数字（0～9）组成。

③ 大小写字母表示不同意义，即代表不同的标识符。

④ 标识符的长度限制与编译器相关，一般只有前32个字符有效，但是编译器不同，允许的长度也不一样。

⑤ 标识符不能使用关键字。

（3）空白符：指示词法记号的开始和结束位置，在程序编译时不起任何作用，可以被完全忽略掉。

（4）分隔符：用于分隔C语言中的词素、语句的符号，可以是空格、回车/换行、逗号等，分

隔符用于构造程序。

1.1.3　变量与数据类型

（1）变量：在程序中，其值是可以被改变的量。变量名必须是合法的标识符。

（2）数据类型：用来确定数据的取值范围和运算方式。本章只介绍 4 种数据类型，即整型（int）、字符型（char）、单精度浮点型（float）和双精度浮点型（double）。可以用 signed（有符号）和 unsigned（无符号）对整型和字符型进行修饰，如 signed int 和 unsigned int。

1.1.4　运算符与表达式

（1）运算符：本章所介绍的运算符为 =(14)、+(4)、-(4)、*(3)、/(3)、%(3)，括号中的数字表示运算符的优先级。

（2）表达式：由运算符、变量或常量组成，如 a = 2 为赋值表达式。

1.1.5　系统函数

本章介绍两个非常重要的系统函数——格式输出函数 printf()和格式输入函数 scanf()。

（1）格式输出函数

它的功能是按照指定的格式向标准输出设备（通常为显示器）输出指定的内容，一般形式为

```
printf("%格式字符串", 变量);
```

本章所涉及的格式字符串有：输出变量为整型用"%d"，输出变量为字符型用"%c"，输出变量为单精度浮点型用"%f"，输出变量为双精度浮点型用"%lf"。

（2）格式输入函数

scanf()函数作用是按指定格式从标准化输入设备（通常指键盘）读入数据，其调用一般形式为

```
scanf("<格式字符串>", <参量表>);
```

scanf()函数的要求与 printf()函数相似，本章所涉及的格式字符串有：输入字符型用"%c"，输入有符号整型使用"%d"，输入单精度浮点型用"%f"等。不过，参量表中的变量前面需要加上一个符号&。&被称为取地址运算符，运算级别为 2。它的含义为把由键盘输入的数据存入参量表中指定地址的内存中，并以回车作为输入结束。

1.1.6　流程图

流程图是表达程序设计思路的有效方式，本章介绍 4 种符号，如图 1-1 所示。

图 1-1　流程图符号

1.1.7　编程风格

（1）添加适当的注释。

（2）格式控制的使用，每个层次（常以一对“{}”为一层次）要有适当的缩进。

（3）要遵循变量和函数的命名规则与标准。

1.2 课后习题指导

1. 填空题

（1）答：表达式的值 = 4；a = 4；b = 10；c = 6。

提示：这是一个复合表达式，要按运算符的优先级别来进行，其中“( )”的优先级最高，先执行(b = 10)和(c = 6)，然后执行%运算，即 b%c = 10%6 = 4，最后把 4 赋给 a。

（2）答：%取余运算符。

2. 选择题

（1）答：D。

（2）答：D。**提示：**注释也属于空白符，在 C 语言编译时，所有空白符都将被忽略。

（3）答：D。**提示：**5/2 = 2，5%2 = 1，所以表达式 = 3.6−2 + 1.2 + 1 = 3.8。

（4）答：A。**提示：**%d 为十进制输出，%o 为八进制输出，注意 n = 032767 本身为八进制赋值方式。

（5）答：B。**提示：**%u 为无符号十进制输出，需要对 x 进行转换。

（6）答：C。**提示：**A 中%不能用于小数，B 中赋值符号左边不能为常量 D 中与 3 相连的等号左边为常量。

（7）答：C。**提示：**整型运算结果为整型，即有 3/2 = 1，结果与 x 相加后，转换为双精度浮点型且 y 也为浮点型，故结果为 2.0。

（8）答：B。**提示：**A 为关键字，不能使用；C 第一个字母不是字母或下划线；D 中含有 +，它不能出现在合法的标识符中。

（9）答：C。

（10）答：D。**提示：**A 以“;”为分隔符，为 3 条语句，每一条都需要加类型声明，即 int b = 5; int c = 5。B 没有对 a、b 进行初始化；C 没有类型声明。

3. 简答题

（1）程序改错是编程的基本功训练项目，是对读者学习效果的一种检验，能够帮助读者认识到编程中易犯的错误，从而避免编程时发生同样的问题。

```
Include "stdio.h"                                  //缺少""且头文件写错
int s;                                             //原题缺少;号
s=52;                                              //原题多了;号
printf("There are s%d weeks in a year.",s);        //原样输出的语句应该包含在""内,且整型
                                                     变量有格式
```

（2）关键字为：int，char。

提示：main 为系统预定义的标识符，function 为普通标识符，= 为运算符。

（3）关于编程风格需要注意的问题。

① 添加适当的注释。

② 格式控制的使用，在每个层次（常以一对“{}”为一层次）要有适当的缩进。

③ 要遵循合适的变量和函数的命名规则与标准。

4. 编程题

初学者会片面地把编程理解为纯粹的代码编写，其实编程更重要的是程序设计思路的体现。

程序设计思路的表达必须在初学时就开始进行训练，否则“贻害无穷”！因此本题先绘制流程图，再编写代码。

（1）输出所要求的图形的流程图如图 1-2 所示。

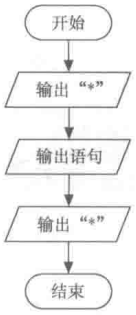


图 1-2　输出所要求的图形

```
#include "stdio.h"
void main(void)
{
    printf("**********************\n");//输出*
    printf("   Very  Good!\n");          //输出语句
    printf("**********************\n");//输出*
}
```

（2）计算三角形面积的流程图如图 1-3 所示。

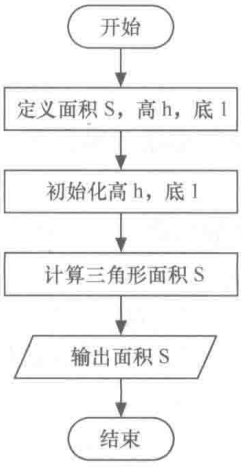


图 1-3　计算三角形面积

```
#include "stdio.h"
void main(void)
{   int nS,nH,nL; //定义变量
    nH=4; nL=6;  //初始化变量
    nS=0.5*4*6; //计算三角形面积
    printf("%d\n",nS);//输出语句
}
```

扩展训练：若 nH = 3，nL = 5，结果会怎样？如何更改？

5. 思考题

（1）计算图形面积的流程图如图 1-4 所示。

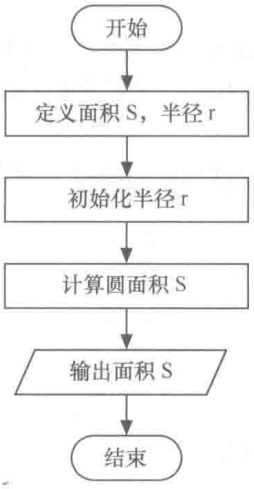


图 1-4　计算圆面积

（2）量出 4mL 水的流程图如图 1-5 所示。

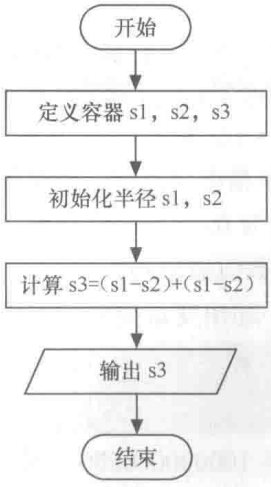


图 1-5　量出 4mL 水

1.3 实验问题解答

1. 如何理解 C 语言中的数据类型？

答：C 语言中的数据类型有两方面的作用。

（1）规定了变量的取值范围，如 int 的取值范围为-2147483647～2147483648。

（2）规定了变量的运算方式，如 int 可进行加、减、乘、除运算，而 char 则不能。

2. 如何理解常量与变量？

答：变量是在程序中可以变化的量，如：

```
int nA=5;    //初始化为 5
nA=10;       //被改变为 10
```

常量是不能被改变的，如上面例子中等号右边的 5、10。在今后的学习中，读者还会遇到其他形式的常量定义方式。

3. 学习运算符时，应注意哪些事项？

答：学习运算符时要注意以下几点。

（1）对运算符两端操作数的数量的要求。本章所遇到的运算符都是双操作数，以后还会遇到单操作数和三操作数。

（2）对运算符两端操作数的类型的要求。如%不能用于非整数运算。

（3）在含有多个运算符时，要注意每个运算符的优先级，如：

```
b = c*d + 1*5 - (f + 8);
```

因为()的优先级别最高，所以先计算（f+8），然后先乘除、后加减，最后是赋值。

（4）注意运算时的类型转换，如：

```
int nA=1/2;       //nA 的值为 0,整型运算
nA=5/3;           //nA 的值为 1,整型运算
float fB=1/2.0;  //fB 的值为 0.5,运算时先将 1 转换为 1.0,再进行除法运算
```

4. 表达式的值与变量的值的区别。

答：表达式的值通常取决于在该表达式中级别最低的（即最后运算的）运算符运算完毕后的值，如：

```
a = b*c + 3*d;          //表达式的值为 a 的值，因为在这个表达式中" = "级别最低
a = (b = 3)+(d=2);    //尽管有 3 个赋值语句，但"( )"改变了它们的运算次序
```

5. 为何 float a = 1/2 为 0？

答：虽然 a 为单精度浮点型，但赋值语句的右边 1/2 仍为整数除法，其运算结果为 0，所以把 0 赋给 a，其值仍为 0。

6. 运算会产生数值溢出吗？

答：当运算时，超出变量类型的取值范围时，就会产生溢出，如：

```
int nA=10000, nB=10000;
nA=nA*nB*nB;
printf("%d",nA );
```

输出结果不会是 100000000000，因为这个值超出了整型的类型。

7. 使用 printf()函数时应注意哪些情况？

答：（1）注意不要把要输出的变量也放到" "中去，如：

```
printf("%d a");
```

a 为要输出的整型变量，正确写法为

```
printf ( " %d ", a ) ;
```

（2）注意控制输出的格式字符串类型与定义的变量类型相匹配，如：

```
int nA;…; printf("%d", nA);          //正确
float fB;…; printf("%f", fB);        //正确
float fC;…; printf("%d", fC);     //错误
```

第 2 章 选择控制结构及其应用

2.1 本章学习辅导

2.1.1 选择控制条件

（1）关系运算符：共有 6 个，分别为>（6）、>=（6）、<（6）、<=（6）、= =（7）、!=（7），括号内的数字为该运算符的优先级，下同。

（2）逻辑运算符：共有 3 个，分别为与&&（11）、或||（12）、非!（2）。

（3）条件运算符：?:（13）。一般形式为

```
表达式 1? 表达式 2:表达式 3;
```

表达式 1 为真（非 0）执行表达式 2，整个表达式值为表达式 2 的值；表达式 1 为假（0），执行表达式 3，整个表达式的值为表达式 3 的值。

（4）条件表达式：由关系运算符和逻辑运算符组成的表达式，称为条件表达式，它可以构成选择控制条件。

（5）隐式类型转换：当不同数据类型混合运算或把低精度数赋给高精度数时，系统会自动地把低精度的数转换为高精度的数，然后再作运算，以确保整个过程不会丢失精度。数据类型由高到低的排列顺序为：char→short→int→unsigned→long→float→double。

（6）强制类型转换：由高精度数据向低精度数据转换的过程。这种转换可能会造成精义的丢失，系统不能自动进行，一般形式为

```
低精度变量 = (低精度数据类型 )高精度数据变量;
```

例如：int nA; float fB=123.34; nA=(int) fB; 则 a=123，nA“丢失”了 fB 的小数部分。赋值表达式本身也具有强制转换的性质，即把赋值符号右边的变量强制转换为赋值符号左边的变量类型，并将转换结果赋给右边的变量，如 nA=fB。

2.1.2 if–else 条件选择控制结构

（1）形式 1：if (条件表达式) ｛语句块 1;｝表达式为真（非 0）时，执行语句块。简单的 if 结构流程图如图 2-1 所示。

（2）形式 2：表达式为真（非 0），执行语句块 1，为假（0），执行语句块 2。if-else 结构流程图如图 2-2 所示。

（3）形式 3：按次序考察表达式的值，决定执行哪一语句块，但最终只能执行一个语句块。if-else-if 流程图如图 2-3 所示。

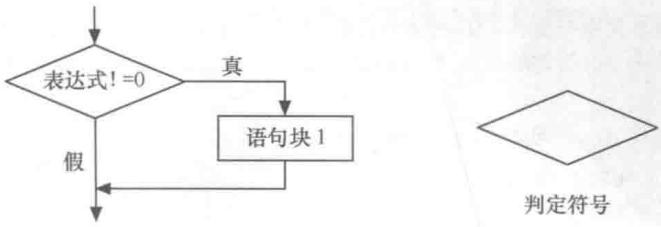


图 2-1　简单的 if 结构流程图

```
if(表达式){
    语句块 1; }
else{
    语句块 2;   }
```

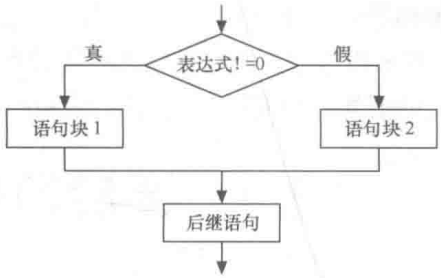


图 2-2　if-else 结构流程图

```
if(表达式 1)
    语句块 1;
else if(表达式 2)
    语句块 2;
else if(表达式 3)
    语句块 3;
    …
else if(表达式 n)
    语句块 n;
else
    语句块 n+1;
    …
```

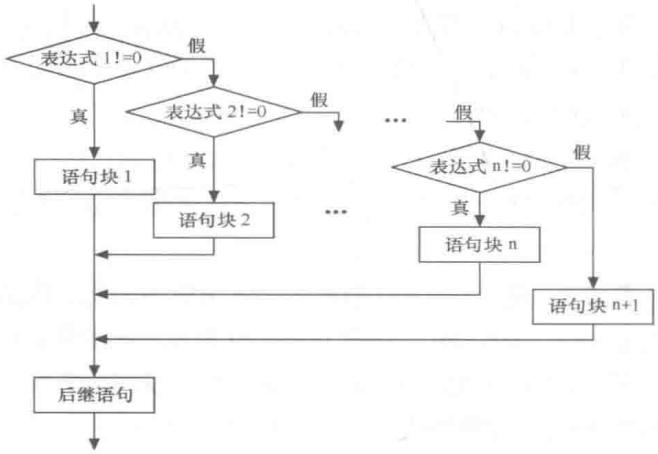


图 2-3　if-else-if 流程图

（4）if 语句的嵌套：将上述三种形式组合起来，可形成 if 语句的嵌套，这时会出现多个 if 和多个 else 重叠的情况。C 语言规定，else 总是与它前面最近的 if 配对。

```
if (表达式)
{
      if (表达式)
      {
            语句;
      }
}
```

```
if (表达式)
{   if (表达式)
    { 语句;}
 }
else
{   if (表达式)
    { 语句;}
}
```

2.1.3　switch 判定结构

当判定条件中的常量或变量只包括字符或整型，且运算关系仅为是否等于（“= =”），则可使