



普通高等教育“十一五”规划教材

# C语言程序设计

(第二版)

刘加海 朱云其 编著



科学出版社

[www.sciencep.com](http://www.sciencep.com)

TP312/3654

2010

普通高等教育“十一五”规划教材

# C 语言程序设计

(第二版)

刘加海 朱云其 编著

科学出版社

北京

## 内 容 简 介

本书系统地介绍了C语言程序的设计思路、设计方法及开发流程,通过大量的程序设计实例及与实例相仿的思考题,能使读者尽快提高C语言程序设计能力。

本书结构合理,概念清楚,由浅入深,实用性强,程序中的函数、循环、数组与基本算法(如查找、排序、删除等)融为一体。全书从字符串输出开始,到菜单的显示与选择、记录的输入与显示、记录的查找、排序、删除、文件存储等,程序设计的思想贯穿全书。

本书是一本技能型、工程型的教材,非常适合于C语言程序员及软件工程技术人员使用,也可以作为计算机类、电子信息类相关专业本、专科学生的教材。

### 图书在版编目(CIP)数据

C语言程序设计(第二版)/刘加海,朱云其 编著. —2版. —北京:科学出版社,2010

(普通高等教育“十一五”规划教材)

ISBN 978-7-03-026309-4

I. C… II. ①刘…②朱… III. C语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2009)第241609号

责任编辑:孙露露 / 责任校对:柏连海

责任印制:吕春珉 / 封面设计:耕者设计工作室

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

数字印刷厂印刷

科学出版社发行 各地新华书店经销

\*

2003年8月第一版 开本:787×1092 1/16

2010年2月第二版 印张:19

2010年2月第十次印刷 字数:425 000

印数:26 001—29 000

定价:28.00元

(如有印装质量问题,我社负责调换<环伟>)

销售部电话 010-62134988 编辑部电话 010-62138978-8212

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

## 第二版前言

C 语言是目前使用最为广泛的计算机高级语言,发展如此迅速,已成为最受欢迎的语言之一。C 语言功能丰富,表达能力强,使用灵活方便,应用面广,目标程序效率高,可移植性好,既具有高级语言的优点,又具有低级语言的许多特性。归纳起来,C 语言具有下列特点。

1) C 语言是中级语言。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元。

2) C 语言是结构式语言。结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言是以函数形式提供给用户的,这些函数可方便地调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

3) C 语言功能齐全。C 语言具有各种各样的数据类型,并引入了指针概念,可使程序效率更高。另外,C 语言也具有强大的图形功能,支持多种显示器和驱动器,而且计算功能、逻辑判断功能也比较强大,可以实现决策目的。

4) C 语言适用范围广。C 语言还有一个突出的优点就是适合于多种操作系统,如 DOS、Windows、UNIX、Linux,也适用于多种机型。

本书共分 12 章,各章主要内容介绍如下。

第 1 章主要讲述了程序的概念,对提出的某个问题,如何分析,解决问题的方法与步骤——即算法是什么,为什么要采用这种算法?掌握程序的顺序、分支和循环这三种结构,正确理解对程序过程的描述、流程图的结构及流程图的画法。对本章中的源程序只作一般性的了解即可,例如了解大致的程序结构,如 main 函数,程序从什么地方开始、到什么地方结束,什么是语句,C 语言中书写的格式及是否区分大小写等。也可以暂时不学习源程序,而把学习的重点放在对问题的描述、解决问题的方法以及程序的调试过程上。能够掌握整型常量的表示方法,变量的定义方法,int、char、float、double 等各种基本数据的输入/输出方法,能够运用算术运算符、关系运算符和逻辑运算符构成各种表达式。

第 2 章主要讲述了 C 语言程序设计中程序的格式化输入与输出,输入/输出中各种不同类型变量的格式控制,尤其是从键盘输入时是输入到变量的地址上,能够输出文本菜单与文本构成的各种图形。

第 3 章主要讲述了 C 语言程序设计中的 while 循环语句。通过本章的学习,能够应用 while 循环解决实际工程问题,掌握 while 循环的程序执行流程,能正确应用 continue 与 break 语句,掌握循环嵌套的程序设计。建议重点掌握 while 语句、for 语句的格式、for 语句中的两个分号“;”及三个表达式的变化形式以及 do-while 语句的格式,while 语句的执行过程,应用 while 语句来编写程序,在程序编写过程中应注意循环变量的变化情况及循环体的模块结构;掌握 break 与 continue 的区别;掌握如何应用 for 语句来

编写程序；初步掌握 for 循环的嵌套程序设计。

第 4 章主要论述了 C 程序设计中的 if 语句的用法。要正确理解 if 语句中表达式的用法，此表达式通常采用的是逻辑表达式、关系表达式等，但有时也采用算术表达式，甚至有时使用赋值表达式。在实际应用中，if 语句通常与循环语句一起使用。switch 语句的用法虽然简单，但使用过程中也容易犯错，建议在学习过程中着重理解 break 语句的使用，注意 case 后面的常量或表达式，注意关系运算表达式、逻辑运算表达式所引起的错误，注意 switch 中表达式的类型与常量的类型必须一致。同时，要注意在 case 分支后是否有 break 语句所引起的程序走向。

第 5 章主要论述了数组在程序设计中的应用。数组是一种常用的数据类型，数组的应用非常广泛，例如数组的操作往往涉及到元素的排序、查找、插入、删除等。数组元素的查找中涉及到顺序查找与二分法查找，数组元素的插入涉及插入到某位置与按序插入。当插入元素时，插入点后面的每一个元素先向后移一个位置，在空出的位置插入此数。在数组元素删除中，首先定位到删除的元素，从删除元素的后一元素开始，依次向前移一个位置。在数组的操作过程中，应注意元素下标值的变化情况。

第 6 章主要论述了函数的基本概念、系统函数的调用、函数的定义及函数的应用。函数有系统函数与自定义函数，系统函数调用时要包含相应的库函数，自定义函数时一定要认真关注此函数的返回值类型、函数名、函数的参数及参数的适用范围、函数调用中的实参与函数定义中的形参的对应关系，希望读者多实践，编写程序上机调试。

第 7 章主要论述了不同存储类型的变量在不同函数、不同文件中生存期与作用范围的不同。理解变量的存储类型与变量的作用范围、生存期之间的关系，掌握自动变量的作用范围，掌握内部静态变量与外部静态变量作用范围及生存期，外部变量在多文件系统中的应用。

第 8 章主要论述了文件的基本概念与应用。在操作系统中，外部存储器中的数据、文档或程序，都是以文件的形式保存，甚至对目录和各种设备的操作也都等同于文件的操作。本章让读者初步掌握 C 程序设计中文件的基本操作，读者在学习过程中应关注打开的是文本文件还是二进制文件，打开操作前文件是否已存在，如何判断文件的结束，以什么样的格式读写文件，用什么函数来读写。

第 9 章主要论述了指针的概念与应用。指针在 C 语言中是非常重要的概念，指针可以指向变量的地址，指针使用灵活，可以使用++、-移动指针来获取指针所指的地址上的内容，因而把指针应用在数组上非常灵活。C 语言利用指针变量统一地处理变量的地址和数组的下标，因而提高了 C 语言的描述能力，并可获得较高的编译质量。指针变量被广泛使用，它和数组、字符串、函数间数据的传递等有着密不可分的联系，学习中应注意：虽然指针变量是指向变量在内存中的存储地址的，而实际上更关心的是指针变量所指的地址上的内容。

第 10 章主要论述了指针和函数的关系。主要包括四个方面的内容：一是指针可以作为函数的参数进行传递；二是某种指针可以指向函数的入口地址，即指向函数指针；三是调用一个函数后，函数的返回值可以是指针，即指针函数；四是字符类型的指针数组可以作为 main 函数的参数，在程序执行时从命令行传入，即所谓的命令行参数。在

本章中涉及的概念较多，要区分各种类型的定义形式与应用方法。

第 11 章主要论述了结构体类型的概念及应用。结构体是一种自定义的数据类型，结构体在增强程序语言功能方面有不可替代的作用，特别是涉及到一些结构化的且有大量数据处理的程序，常常要借助结构体变量特别是结构体数组变量的使用。本章从结构体类型的定义、初始化、赋值开始，讲解结构体变量的输入/输出、增加、显示、查找、删除、排序，利用结构体数组与文件之间的操作实现了一个结构化的通讯录程序。

第 12 章主要论述了链表的定义、栈的基本操作的实现、队列的基本操作的实现。链表属于动态空间分配，主要优点是可以根据程序数据的存储需要随时扩充或缩小数据结点，这样不会浪费存储空间，不过链表中相邻的结点其存储地址是不相邻的。链表的建立过程是用指针把前后结点连接起来，程序设计中应注意是队列结构还是栈结构，队列是先进先出的存储结构，而栈是先进后出的存储结构。队列的建立过程中头指针不动，而尾指针不断指向新产生的结点；栈的建立过程中尾指针不动，而头指针不断指向新产生的结点。

本书在讨论以上这些主题时，首先给出实例，通过实例介绍程序设计的方法，通过大量的实例与清晰的程序流程使读者迅速掌握相关知识、编程技能与技巧，然后给出相关联的思考题，通过思考题迅速提高读者对程序设计概念的理解；通过给出的相应的上机实践，提高读者的程序设计能力和掌握程序设计技巧。

本书的作者对 C 语言教学有着丰富的实践与教学经验，本书由浙江大学城市学院刘加海、浙江工商大学朱云其、辽宁工程技术大学黄金波、河南工业大学李周其编著。

本书可提供相应的教学大纲、教学课件、习题答案及所有的源程序，可到科学出版社网站（[www.abook.cn](http://www.abook.cn)）下载。由于时间仓促及作者水平有限，书中难免存在疏漏和不妥之处，敬请广大读者批评和指正，批评与建议请发到邮件 [Ljhqyyq@yahoo.com.cn](mailto:Ljhqyyq@yahoo.com.cn)，非常感谢。

## 第一版前言

C 语言是一种优秀的结构化程序设计语言，它结构严谨、数据类型完整、语句简练灵活、运算符丰富。自从应用 C 语言编写 UNIX 操作系统获得成功以来，它一直受到计算机业内人士的好评。即使在面向对象的语言广泛使用的今天，C 语言仍然是高校计算机专业的一门专业基础课。

几乎每一所高校都开设 C 语言这门课，然而真正易教易学的教学却不多，这是笔者在浙江大学教授近 100 轮次的 C 语言教学课后的深刻体会。目前大多数 C 语言教材先讲授数组，在课程快结束时才讲指针。很多学生对指针的概念感到难以理解，对于它的灵活应用更是困难。从 1996 年开始，笔者根据多年的教学经验，尝试在每一章就讲授指针与数组，把数组和指针有机地结合起来，并且全书贯穿指针与数组的概念，使学生一开始就对 C 语言有总体上的清晰认识，这种方法在后来的教学中取得了很大的成功，从心理学的角度来看这也是完全符合人类的心理学习规律。近来笔者对这种教学方法进行了反复的研究，并进行多次改进和进一步完善，结合高职高专的教学特点，编著了这本《C 语言程序设计》。

我们不敢说哪种方法最好，但必须尝试一种教学方法，它不但要符合此课程的教学特点，使学生在在学习过程中感到轻松，更要符合人们学习的心理学习规律，花更少的时间，能更好地掌握知识，希望这本书能让广大师生达到此目的。

尽管笔者修改再三，但无论书的结构还是书的内容都难免有不足的地方，真切地希望广大师生批评指正，以便在下一版中订正。

本书由刘加海担任主编，齐幼菊、孔美云、周文革任副主编，参加本书编写的还有戚海燕、朱光忠、张翔、期进、郑正建、徐寿芳，在这里对他们的辛苦工作表示感谢。

# 目 录

|                           |    |
|---------------------------|----|
| 第 1 章 C 语言程序的基本构成 .....   | 1  |
| 1.1 程序的概念 .....           | 2  |
| 1.2 算法与流程图 .....          | 3  |
| 1.3 结构化程序的结构 .....        | 4  |
| 1.3.1 顺序结构 .....          | 4  |
| 1.3.2 分支结构 .....          | 7  |
| 1.3.3 循环结构 .....          | 9  |
| 1.4 C 程序的构成 .....         | 11 |
| 1.5 C 函数 .....            | 14 |
| 1.6 C 语句 .....            | 15 |
| 1.7 保留字与标识符 .....         | 15 |
| 1.7.1 保留字 .....           | 15 |
| 1.7.2 标识符 .....           | 16 |
| 1.8 常量 .....              | 16 |
| 1.8.1 整型常量 .....          | 16 |
| 1.8.2 实型常量 .....          | 17 |
| 1.8.3 字符常量 .....          | 17 |
| 1.9 变量 .....              | 18 |
| 1.9.1 变量的数据类型及其定义 .....   | 18 |
| 1.9.2 整型变量 .....          | 19 |
| 1.9.3 实型变量 .....          | 19 |
| 1.9.4 字符变量与字符串 .....      | 20 |
| 1.10 运算符 .....            | 23 |
| 1.10.1 算术运算符 .....        | 23 |
| 1.10.2 关系运算符 .....        | 24 |
| 1.10.3 逻辑运算符 .....        | 25 |
| 1.10.4 位运算符 .....         | 26 |
| 1.11 编辑 C 程序时应注意的问题 ..... | 28 |
| 习题 .....                  | 28 |
| 第 2 章 输入/输出程序设计 .....     | 31 |
| 2.1 字符串的输入与输出 .....       | 32 |
| 2.1.1 字符串的输出 .....        | 32 |
| 2.1.2 字符串的输入 .....        | 33 |
| 2.2 整型数的输入与输出 .....       | 35 |

|                                 |           |
|---------------------------------|-----------|
| 2.2.1 整型数的输出 .....              | 35        |
| 2.2.2 整型数的输入 .....              | 36        |
| 2.3 字符的输入与输出 .....              | 39        |
| 2.4 实型数的输入与输出 .....             | 40        |
| 习题 .....                        | 42        |
| <b>第 3 章 while 循环程序设计</b> ..... | <b>43</b> |
| 3.1 循环的概念 .....                 | 44        |
| 3.2 while 循环 .....              | 44        |
| 3.3 do-while 循环 .....           | 50        |
| 3.4 for 循环 .....                | 53        |
| 3.5 break 语句和 continue 语句 ..... | 57        |
| 3.6 循环嵌套 .....                  | 60        |
| 习题 .....                        | 64        |
| <b>第 4 章 分支结构的程序设计</b> .....    | <b>68</b> |
| 4.1 if 语句的应用 .....              | 69        |
| 4.2 if-else 语句的应用 .....         | 72        |
| 4.3 if-else if 语句的应用 .....      | 75        |
| 4.4 if 嵌套语句的应用 .....            | 78        |
| 4.5 switch 的应用 .....            | 79        |
| 习题 .....                        | 88        |
| <b>第 5 章 数组及其应用</b> .....       | <b>92</b> |
| 5.1 一维数组 .....                  | 93        |
| 5.1.1 一维数组的定义 .....             | 93        |
| 5.1.2 一维数组的引用方法 .....           | 94        |
| 5.1.3 一维数组的初始化与赋值 .....         | 94        |
| 5.1.4 字符数组 .....                | 96        |
| 5.2 二维数组 .....                  | 97        |
| 5.2.1 二维数组的定义 .....             | 97        |
| 5.2.2 二维数组的表示、初始化与赋值 .....      | 98        |
| 5.3 数组元素的查找 .....               | 99        |
| 5.3.1 无序数组的查找 .....             | 100       |
| 5.3.2 有序数组的查找 .....             | 103       |
| 5.4 数组元素的插入 .....               | 105       |
| 5.4.1 绝对位置的插入 .....             | 105       |
| 5.4.2 有序数据的插入 .....             | 107       |
| 5.5 数组元素的删除 .....               | 110       |
| 5.6 数组的排序 .....                 | 111       |
| 5.6.1 冒泡排序 .....                | 111       |

|                                 |            |
|---------------------------------|------------|
| 5.6.2 选择排序 .....                | 114        |
| 习题 .....                        | 117        |
| <b>第 6 章 函数及其应用 .....</b>       | <b>120</b> |
| 6.1 函数的基本概念 .....               | 121        |
| 6.2 函数的参数 .....                 | 122        |
| 6.3 系统函数的应用 .....               | 124        |
| 6.3.1 数学函数 .....                | 124        |
| 6.3.2 字符串处理函数 .....             | 125        |
| 6.3.3 字符操作函数与转换函数 .....         | 127        |
| 6.4 自定义函数 .....                 | 128        |
| 6.5 函数的应用 .....                 | 130        |
| 6.6 函数的嵌套与递归调用 .....            | 140        |
| 6.6.1 函数的嵌套调用 .....             | 140        |
| 6.6.2 函数递归调用 .....              | 141        |
| 习题 .....                        | 144        |
| <b>第 7 章 变量存储类型与多文件系统 .....</b> | <b>148</b> |
| 7.1 变量的存储类型 .....               | 149        |
| 7.2 auto 存储类型的变量与作用范围 .....     | 150        |
| 7.3 static 存储类型的变量与作用范围 .....   | 152        |
| 7.4 register 存储类型的变量与作用范围 ..... | 154        |
| 7.5 extern 存储类型的变量与作用范围 .....   | 155        |
| 7.6 多文件系统中的函数调用 .....           | 157        |
| 7.7 外部变量在多文件系统中的应用 .....        | 161        |
| 习题 .....                        | 162        |
| <b>第 8 章 文件的操作及其应用 .....</b>    | <b>164</b> |
| 8.1 文件概述 .....                  | 165        |
| 8.2 文件打开与关闭函数 .....             | 166        |
| 8.2.1 文件打开函数 fopen .....        | 166        |
| 8.2.2 文件关闭函数 fclose .....       | 167        |
| 8.3 文件中字符的读/写函数 .....           | 168        |
| 8.4 文件中格式化读/写函数 .....           | 170        |
| 8.5 文件中数据块读/写函数 .....           | 173        |
| 8.6 文件中位置定位函数 .....             | 176        |
| 习题 .....                        | 178        |
| <b>第 9 章 指针与数组 .....</b>        | <b>182</b> |
| 9.1 指针的基本概念 .....               | 183        |
| 9.2 相同类型指针间的基本运算 .....          | 185        |
| 9.3 一维数组与指针 .....               | 187        |
| 9.4 字符串与字符指针变量 .....            | 191        |



|                       |     |
|-----------------------|-----|
| 9.4.1 字符数组与字符串        | 191 |
| 9.4.2 指针变量与字符串        | 193 |
| 9.5 指向二维数组的指针         | 194 |
| 9.5.1 二维数组的表示及应用      | 194 |
| 9.5.2 指向二维数组的数组指针     | 196 |
| 9.6 指针数组              | 198 |
| 9.7 二级指针              | 201 |
| 习题                    | 203 |
| 第 10 章 指针与函数          | 210 |
| 10.1 指针与函数参数          | 211 |
| 10.2 指向函数的指针          | 218 |
| 10.3 返回值为指针的函数        | 222 |
| 10.4 命令行参数            | 225 |
| 习题                    | 228 |
| 第 11 章 结构体及其应用        | 231 |
| 11.1 结构体的基本概念         | 232 |
| 11.1.1 结构体类型定义        | 232 |
| 11.1.2 结构体变量的定义       | 232 |
| 11.1.3 结构体变量占据的内存空间   | 233 |
| 11.1.4 结构体变量对结构体成员的引用 | 233 |
| 11.1.5 结构体变量的赋值       | 233 |
| 11.2 结构体类型的数组         | 236 |
| 11.3 结构体变量的输入/输出与存储   | 238 |
| 11.4 结构体变量查找与删除       | 243 |
| 11.5 结构体变量的排序         | 247 |
| 习题                    | 252 |
| 第 12 章 链表及其应用         | 256 |
| 12.1 链表的定义            | 257 |
| 12.2 堆栈               | 258 |
| 12.3 队列               | 260 |
| 12.4 链表的插入            | 263 |
| 12.5 链表的删除            | 265 |
| 12.6 链表的应用            | 267 |
| 习题                    | 275 |
| 附录                    | 276 |
| 附录一 ASCII 表           | 276 |
| 附录二 运算符及其优先级汇总表       | 277 |
| 附录三 C 语言的保留字          | 278 |
| 附录四 C 语言部分常用库函数       | 278 |

|                              |     |
|------------------------------|-----|
| 附录五 Linux 环境下 C 程序编辑、编译..... | 285 |
| 附录六 Visual C++ 6.0 上机步骤..... | 285 |
| 参考文献 .....                   | 287 |

# 第 1 章

## C 语言程序的基本构成

### ● 本章重点

- 程序的概念
- 结构化程序的结构
- 算法的概念
- 流程图的表示
- 程序的框架

### ● 本章难点

- C 语言程序的结构
- 源文件、目标文件、可执行文件的概念
- 算法的理解
- 流程图的画法

### ● 本章学习建议

把握针对提出的某个问题，如何分析，解决问题的方法与步骤——即算法是什么，为什么要采用这种算法？掌握程序的顺序、分支和循环这三种结构，正确理解对程序过程的描述、流程图的结构及流程图的画法。对本章中的源程序只作一般性的了解即可，例如了解大致的程序结构，如 main 函数，程序从什么地方开始、什么地方结束，什么是语句，C 语言中书写的格式及是否区分大小写等。也可以暂时不学习源程序，而把学习的重点放在对问题的描述、解决问题的方法以及程序的调试过程上。

## 1.1 程序的概念

程序是指人们将需要计算机做的工作写成一定形式的指令，并把它们存储在计算机的内部存储器中。当人们给出命令之后，计算机就按指令操作顺序自动进行，这种可以连续执行的一条条指令的集合被称为“程序”。目前，正在使用的计算机程序设计语言有上百种，有些语言是面向机器的，如二进制语言，而多数是面向问题的语言。面向问题的语言被称为计算机的“高级语言”，如 C、C++ 语言等。目前的程序设计一般可分为非结构化程序设计、结构化程序设计和面向对象程序设计，C 语言是结构化程序设计语言，而 C++ 是面向对象的程序设计语言。这些语言都是用接近人们习惯的自然语言和数学语言作为语言的表达形式，人们学习和操作起来十分方便。

程序设计的过程一般包括：

- 1) 问题的提出、要求及所采用的数据结构。
- 2) 算法的确定、程序的编制。
- 3) 程序的调试及修改。
- 4) 整理并写出文档资料。

程序是为了某方面的应用而设计的，用 C 语句所写的序列称为 C 源程序，它的后缀为 .c。C 源程序经过编译 (compile) 后生成一个目标文件 (.obj)，链接库函数后生成可执行文件 (.exe)。因而，C 程序的设计过程如图 1.1 所示。

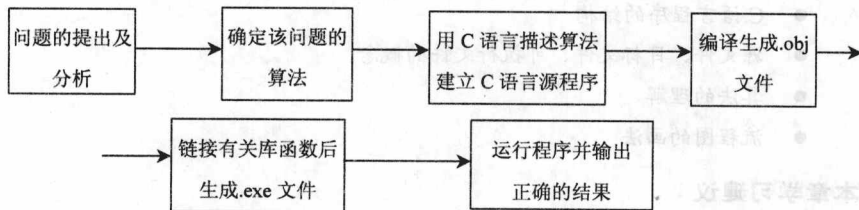


图 1.1 C 程序的设计过程

结构化程序设计由三种结构组成：顺序结构、选择结构和循环结构。

### (1) 顺序结构

顺序结构的程序是一条语句接一条语句顺序往下执行。例如，在图 1.2 中，先执行语句 1，然后执行语句 2，最后执行语句 3。顺序结构的程序是最简单的程序。

### (2) 分支结构

在程序执行过程中，程序的流程可由多路分支组成，根据不同的条件去执行不同的任务。例如，在图 1.3 中程序执行到条件表达式时，首先判断条件是否为“真”，如为真，则执行语句 1，否则执行语句 2。

### (3) 循环结构

在程序中需要根据某项条件重复地执行某项任务若干次或直到满足或不满足某条件为止，这就构成循环结构。例如，在图 1.4 中程序执行到循环条件表达式时，首先判断循环条件是否为“真”，如循环条件为“真”，则执行循环语句，循环语句执行完成后，

又转去判断循环条件，如循环条件仍为真，再次执行循环语句，如此反复，直到循环条件为假，退出此循环。



图 1.2 顺序结构的程序段

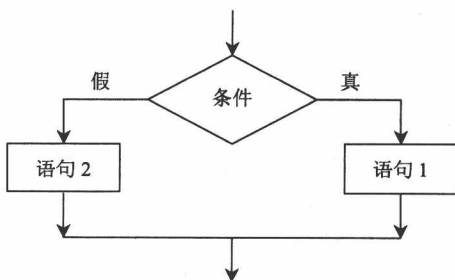


图 1.3 分支结构的程序段

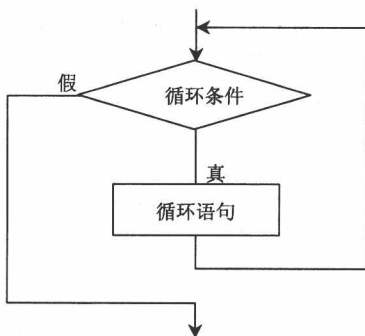


图 1.4 循环结构的程序段

### 思考

某同学起床后，先穿衣、洗脸、吃饭，然后看看课表，根据课表安排带上所需的课本，去教室，上午上五节课，先上一节课，还没到五节课，再上一节课，还没到五节课，再上一节课……直到五节课上完，早上的学习任务结束。根据此段描述，若用程序流程来分析，经历了哪些程序结构？

## 1.2 算法与流程图

### 1. 算法

算法是指解决问题所使用的一系列合乎逻辑的解题步骤，可以把算法定义为解一类确定问题的任意一种特殊的方法。在计算机科学中，算法要用计算机算法语言描述，算法代表用计算机解一类问题的精确、有效的方法。算法+数据结构=程序，求解一个给定的可计算或可解的问题，不同的人可以编写出不同的程序。

制定一个算法，一般要经过设计、确认、分析、编码、测试、调试、计时等阶段。要解决一个复杂的问题，需要仔细的分析与研究此问题，并进行精确的逻辑分析，找到解

决问题的方法，再用计算机实现。例如，判断一个整数的奇偶性，大致可分成以下几步：

- 1) 计算机读入该数。
- 2) 将该数除以 2。
- 3) 判断余数是否为 0。
- 4) 如果为 0，则该数为偶数。
- 5) 否则为奇数。

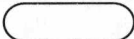
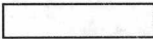
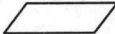
这些解决数的奇偶性问题的一系列步骤就是算法，并且可以用 C 语言编写成程序。

## 2. 流程图

流程图是一种用图解方式来说明解决一个方案所需完成的一系列操作。

流程图比文字更容易理解，更加直观，流程图中的符号及含义如表 1.1 所示。

表 1.1 流程图的符号及含义

| 常用基本组件                                                                              | 含 义      |
|-------------------------------------------------------------------------------------|----------|
|    | 程序的入口和出口 |
|    | 加工、处理    |
|    | 条件判断     |
|    | 控制流      |
|   | 连接符      |
|  | 输入/输出    |

## 1.3 结构化程序的结构

下面通过实例向大家讲解顺序结构、选择结构和循环结构的程序设计方法。

### 1.3.1 顺序结构

顺序结构的程序是一条语句接一条语句顺序地往下执行，直到程序结束。

【例 1-1】 设计一个程序，显示下列图形：

```

*****
C 语言的学习要努力!
*****

```

**分析** 这是一个典型的顺序执行的程序设计，输出三个字符串，输出字符串应用函数 `printf` 来完成，程序设计具体的过程描述如下：

- 开始
- 显示：\*\*\*\*\*
- 换行后再显示：C 语言的学习要努力！
- 换行后再显示：\*\*\*\*\*
- 程序结束

此程序设计用流程图表示如图 1.5 所示。

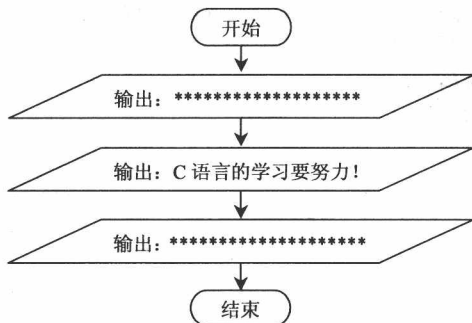


图 1.5 程序设计流程图

用 C 语言写成的源程序如下：

```
#include<stdio.h>                                     /*包含标准输入/输出库预处理命令*/
int main()                                              /*主函数开始*/
{                                                       /*程序从此开始执行*/
    printf("*****\n");                                /*输出一行星号*/
    printf("C 语言的学习要努力! \n");                 /*输出一行字符*/
    printf("*****\n");                                /*输出一行星号*/
    return 0;                                           /*与 main 函数中的 int 相对应*/
}                                                       /*程序结束*/
```

注意：C 语言程序设计中往往要用到输入与输出函数 `scanf`、`printf`，因而要包含预处理命令 `#include<stdio.h>`。程序从 `main` 函数开始执行，一个能运行的 C 语言程序必须要有一个 `main` 函数，并且只能有一个 `main` 函数，`printf` 为系统输出函数，语句后要有“;”号，细节问题请读者通过第 2 章来学习。



### 上机操作与练习 1

步骤 1：在 Visual C++ 环境下新建文件，可单击菜单“开始→程序→Microsoft Visual C++ 6.0→Microsoft Visual C++ 6.0”。如果出现每日提示，如图 1.6 所示，即单击“关闭”按钮。

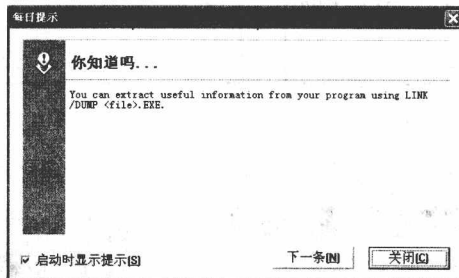


图 1.6 “每日提示”对话框

步骤 2：单击菜单“文件→新建→文件→C++ Source File”，输入文件名 1-1.c，如