

## <<C语言程序设计导学>>

### 图书基本信息

书名：<<C语言程序设计导学>>

13位ISBN编号：9787030350787

10位ISBN编号：7030350782

出版时间：2012-7

出版时间：科学出版社

作者：杜友福

页数：186

字数：288000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<C语言程序设计导学>>

### 内容概要

《C语言程序设计导学（第三版）》是《C语言程序设计导学》的第三版，为C语言程序设计课程的学习指导和实验教材，既可与《C语言程序设计（第三版）》配套使用，也可独立使用。

《C语言程序设计导学（第三版）》分为学习指导和实验指导两部分，学习指导部分分为13章，每章包括学习要求、内容要点、典型例题分析三个方面的内容；实验指导部分根据C语言的教学需要，设计了11个实验。

学习者通过学习与训练，能够进一步掌握C语言的基本概念、C程序设计的基本方法，并通过上机实验强化程序设计与调试的基本功，理解和掌握C语言的基本知识，达到学好和用好C语言的目的。

《C语言程序设计导学（第三版）》特别适合于本、专科非计算机专业的初学者，也可供计算机等级考试者和其他各类学习者使用参考。

## <<C语言程序设计导学>>

作者简介

杜友福

## <<C语言程序设计导学>>

### 书籍目录

学习指导第1章 C语言程序设计概述1.1 学习要求1.2 内容要点1.3 典型例题分析第2章 数据类型、运算符与表达式2.1 学习要求2.2 内容要点2.3 典型例题分析第3章 顺序结构程序设计3.1 学习要求3.2 内容要点3.3 典型例题分析第4章 选择结构程序设计4.1 学习要求4.2 内容要点4.3 典型例题分析第5章 循环结构程序设计5.1 学习要求5.2 内容要点5.3 典型例题分析第6章 数组6.1 学习要求6.2 内容要点6.3 典型例题分析第7章 函数7.1 学习要求7.2 内容要点7.3 典型例题分析第8章 用户标识符的作用域和存储类8.1 学习要求8.2 内容要点8.3 典型例题分析第9章 编译预处理9.1 学习要求9.2 内容要点9.3 典型例题分析第10章 指针10.1 学习要求10.2 内容要点10.3 典型例题分析第11章 结构体、共用体和用户定义类型11.1 学习要求11.2 内容要点11.3 典型例题分析第12章 位运算12.1 学习要求12.2 内容要点12.3 典型例题分析第13章 文件13.1 学习要求13.2 内容要点13.3 典型例题分析实验指导实验一 顺序结构程序设计实验二 选择结构程序设计实验三 循环结构程序设计(一)实验四 循环结构程序设计(二)实验五 数组(一)实验六 数组(二)实验七 函数(一)实验八 函数(二)实验九 指针实验十 结构体实验十一 文件附录 在Visual C++ 6.0集成开发环境下运行C程序

## &lt;&lt;C语言程序设计导学&gt;&gt;

## 章节摘录

1.1 学习要求掌握程序及程序设计的基本概念、程序的运行过程、算法及其特点、算法的描述及其表示方法、C语言程序的基本结构。

熟悉设计程序时应遵循的基本原则。

了解利用计算机解决实际问题的过程、结构化程序设计方法，并在以后的程序设计中应用和逐步熟练掌握。

基本要求：能将日常生活中解决一些简单问题的步骤用算法描述，掌握N唱S图的算法表示方法。

1.2 内容要点【知识点】(1) 程序就是用计算机能够识别的语言所描述的解决某个特定问题的方法和步骤，是由一组相关的指令组成的。

程序 = 数据结构 + 算法 (2) 程序设计就是利用计算机解决问题的全过程。

通常是对问题进行分析并建立数学模型；在此基础上设计出相应的数据结构和算法；然后用某种程序设计语言编写相应的程序代码；最后调试程序，使之运行后能产生预期的结果。

(3) 程序设计语言可分为机器语言、汇编语言、高级语言和面向对象的高级语言四类。

(4) 高级语言程序通常要经过编辑、编译后方能在计算机上运行，少数语言以解释方式执行。

(5) 设计程序时应遵循的基本原则包括：正确性、可靠性、简明性、有效性、可维护性和可移植性。

(6) 计算机算法是为计算机解题设计的有明确意义的运算步骤的有限集合。

算法应具有有穷性、确定性、有效性、有输入输出等特点。

(7) 算法可以用自然语言、伪代码、传统流程图、结构化流程图等方法表示。

(8) 结构化程序设计方法要求在设计程序时只使用顺序、选择、循环三种基本结构。

这三种基本结构有以下共同特点：只有一个入口、只有一个出口、结构内的每一部分都有机会被执行到、结构内不存在“死循环”(无终止的循环)。

(9) 结构化程序设计方法的基本思路是，把一个复杂问题的求解过程分阶段进行，每个阶段处理的问题都控制在人们容易理解和处理的范围内。

结构化程序设计方法的基本原则是：自顶向下、逐步细化、模块化设计、结构化编码。

(10) 函数是C程序的基本单位。

一个函数由函数的首部和函数体两个部分组成。

函数的首部描述该函数的外观(即该函数的名字，该函数需要外部提供哪些信息，执行完毕后能为外部提供哪些结果等)，函数体则定义该函数的内涵(即该函数将对什么数据执行何种操作)。

(11) 一个C源程序至少包含一个函数(即main函数)。

一个C程序总是从main函数开始执行并在该函数结束。

main函数可以放在程序的任意位置。

(12) C程序书写格式自由，一行内可以写多个语句，一个语句也可以写在多行上。

分号是C语句的必要组成部分。

(13) C程序的输入和输出操作是通过调用库函数scanf和printf来完成的。

(14) C程序中可以用“/\*&hellip;&hellip;\*/”对程序的任何部分作注释；也可以用“//&hellip;&hellip;”作为一个单独的注释行。

注释对程序的执行没有任何影响。

【重点】(1) 算法的特点及简单算法的设计。

(2) C程序的基本结构。

【难点】(1) 算法的设计思想。

(2) 用传统流程图和结构化流程图表示算法。

【典型错误分析】1. 算法设计中的逻辑错误对初学者来说，最常见的逻辑错误包括：忘记对计数器、总数等变量进行初始化；选用的用于控制输入循环结束的标记值也是一个合法的输入数据值；一个作为除数的变量的值为0等。

在用C语言编程实现算法后，编译程序往往不能发现这类错误，但程序运行时的结果可能是不正确的

## &lt;&lt;C语言程序设计导学&gt;&gt;

排除源程序中的错误（包括语法错误和逻辑错误）是进行程序设计所必须具备的能力，是对所学知识的综合检验过程，需要清醒的头脑，严谨细致的工作作风，扎实的基础知识以及逻辑分析能力。

对于初学者，调试程序时应注意以下两点：（1）C 编译程序尽管能够指出大多数程序的语法错误，但不可能完全准确地指出错误发生的源程序行，有时应当综观上下文、参考出错信息进行综合分析判断，不要死抠所报的出错行号。

（2）程序中一处发生的错误会产生若干连带的错误信息，即若干条错误信息可能只有第一条是真正的错误，改掉这一错误，所有其他出错信息将随之消失。

因此查错纠错应从最前面的出错点开始，纠正某个错误后，若不能发现其他错误，则应再次编译源程序，按照新的提示信息查找错误。

2.分号使用错误分号是C 语句的组成部分，初学者往往容易遗漏分号而造成语句的不完整，导致编译程序对C 语句无法正确理解。

3.注释使用错误正确的注释形式是/\*注释内容\*/。

初学者容易犯以下错误：忘记了注释的结束符\*/，或者注释用\*/开头，或用\*/结束，或者在注释符号中插入了空格，即写成/\*&hellip;&hellip;\*/的形式。

C 编译程序对于注释部分（起始符/\*与结束符\*/之间）是不进行任何处理的，也就是说，编译时，一旦遇到/\*符号就对其后的任何符号不予处理，直到遇到\*/符号为止。

而对于其他部分则作为C 的语句进行处理。

因此注释符号使用错误会导致编译程序对程序的错误翻译，从而出现语法错误。

1.3 典型例题分析一、选择题例1.1 C 编译程序是。

A.C 程序的机器语言版本 B.一组机器语言指令 C.将C 源程序翻译成目标程序的程序 D.由制造厂家提供的一套应用软件解 本题的关键是掌握编译程序的作用。

C 语言编写的源程序要执行，必须经过以下步骤：用编辑程序将源程序输入并存放在计算机的存储设备中，然后通过C 编译程序将源程序翻译成计算机能识别的目标程序，最后通过连接装配程序将目标指令模块组合成完整的可执行程序才能在计算机上运行。

C 编译程序是系统软件的一部分，并非应用软件。

C 程序的机器语言版本称为目标程序。

因此选项A、D 均不正确。

虽说能够运行的程序都是一组机器语言指令，但选项B 是笼统的，不准确的。

因此，正确答案为选项C。

例1.2 以下叙述中正确的是。

A.C 程序由主函数组成 B.C 程序由函数组成 C.C 程序由函数和过程组成 D.C 程序中的注释由/\*开头，由\*/结束解 本题的关键是掌握C 程序的基本构成。

在初学阶段，由于程序较小，可能只需main（）函数即可，但解决规模较大的实际问题的最好办法是把程序分成较小的模块，每一模块完成一个独立的功能，这样便于管理和调试。

C 语言的程序模块就是函数，因此读者应该明确且牢记：C 程序是由函数组成的。

每个C 语言的源程序可以包含多个函数，但必须有一个且只能有一个主函数。

因此，选项A 是片面的。

在C 程序中不存在“过程”这一程序结构，故选项C 也是错误的。

程序设计中可以对源程序进行必要的注释，以便于阅读和维护。

C 程序中的注释行由/\*开头，由\*/结束，“/\*”和“\*/”必须紧接着，在这两个字符之间不得插入空格。

在选项D 中“/\*”和“\*/”之间都插有空格，故选项D 也是错误的。

因此，正确答案为选项B。

例1.3 C 函数由构成。

A.函数首部和函数体 B.执行语句和非执行语句 C.自变量和因变量 D.定义域和值域解 本题的关键是掌握C 函数的组成。

## &lt;&lt;C语言程序设计导学&gt;&gt;

从数学角度来说，一个函数有自变量和因变量，也有其定义域和值域，但从C语言的角度来说，函数分为函数首部和函数体两个部分，函数的首部描述该函数的外观，函数体则定义该函数的内涵，各种语句用来在函数体中定义各种数据及对这些数据的操作。

因此，正确答案为选项A。

例1.4 一个C程序的执行是。

A.从main ( )函数开始，直到main ( )函数结束  
B.从第一个函数开始，直到最后一个函数结束  
C.从第一个语句开始，直到最后一个语句结束  
D.从main ( )函数开始，直到最后一个函数结束解 本题的关键是掌握C程序的执行过程。

C程序中必须要有一个而且也只能有一个main ( )函数，不论main ( )函数处于程序中的什么位置，C程序执行时总是从main ( )函数开始。

C程序的执行与函数位置无关。

故选项B是不正确的。

C程序中除了顺序结构的语句执行与排列顺序一致外，选择结构和循环结构都会改变语句的执行顺序。

因而选项C是不正确的。

如果程序中有多个函数，其他函数要通过调用才能执行。

而且被调用函数在执行完毕后，必须返回到调用函数处。

故选项D也不正确。

因此，正确答案为选项A。

二、填空题例1.5 C语言源程序文件的后缀是，经过编译后生成文件的后缀是，经过连接后生成文件的后缀是。

解 文件的后缀即文件的扩展名，主要用来表示该文件的类型。

一般情况下，C语言源程序文件的扩展名为C；C语言源程序经过编译生成目标程序，其扩展名为OBJ；目标程序必须经过连接生成可执行程序才能运行，其扩展名为EXE。

因此，本题答案：C，OBJ，EXE。

例1.6 在事先不知道需要连续输入多少个数据的情况下，可用一个值控制输入循环的终止。

解 这是一道关于算法设计方面的题。

算法设计中，经常会在某种条件下重复执行某些操作，这种情况称为循环。

例如，要连续输入多个数据，就可以用循环来实现，循环次数有时是固定数，有时是通过条件来控制的。

就本题来说，由于不知道需要连续输入多少个数据，此时通常采用的办法是设置一个标记值（flag）来控制循环结束，当输入值为标记值时表示所有数据已输入完毕。

显然该标记值应该小心选择，其值不应该在合法输入数据的范围内。

类似的问题在循环中经常会遇到。

因此，本题答案：标记。

例1.7 结构化程序设计方法使用的三种基本结构是、和。

解 这是一道基本概念题。

结构化程序设计方法要求程序中只使用顺序、选择、循环三种基本结构。

这三种基本结构构成的算法不存在无规律的转向，属于结构化的算法，而且已经证明，由以上三种基本结构构成的算法可以解决任何复杂的问题。

因此，本题答案：顺序、选择、循环。

三、阅读题例1.8 下述算法用于求 $1 * 2 * 3 * 4 * 5$ 。

设两个变量，一个变量p代表被乘数（同时用于存放每一步骤的当前所得乘积），一个变量i代表乘数。

算法如下：S1：使 $p = 1$  S2：使 $i = 1$  S3：使 $p * i$ ，乘积仍放在变量p中，可表示为 $p * i \&arr;$  S4：使i的值加1，即 $i + 1 \&arr;$  S5：如果i不大于5，返回重新执行步骤S3以及其后的步骤S4和S5，否则，算法结束。

## &lt;&lt;C语言程序设计导学&gt;&gt;

最后得到p 的值就是5 !

的值如何改动使其成为求 $1 * 3 * 5 * 7 * 9 * 11$ 。

如果要求 $1 + 2 + \dots + 100$  , 应如何修改 ?

解 首先分析一下上述算法中每一步的作用。

步骤S1、S2 两步是对变量赋初值, 即设置初始乘积为1 , 第一个乘数为1。

S3 是进行乘法运算, 并将结果存放到当前所得乘积p 中。

S4 通过改变i 的值得到下一个乘数。

S5 判断是否所有的运算已执行完毕。

求 $1 * 3 * 5 * 7 * 9 * 11$  与求 $1 * 2 * 3 * 4 * 5$  的差别主要在于, 下一个乘数比前一个乘数多2 , 且最后一个乘数是11。

通过以上分析可知, 只需作很少的改动即可: S1 :  $1 \rightarrow p$ ; S2 :  $1 \rightarrow i$ ; S3 :  $p * i \rightarrow p$ ; S4 :  $i + 2$

$\rightarrow i$ ; S5 : 若 $i \leq 11$  , 返回S3 ; 否则, 结束可以看出, 用这种方法表示的算法具有通用性、灵活性

。S3 到S5 组成一个循环, 在实现算法时, 要反复多次执行S3、S4、S5 等步骤, 直到某一时刻, 执行S5 步骤时经过判断, 乘数i 已超过规定的数值而不返回S3 步骤为止。

此时算法结束, 变量p 的值就是所求结果。

如果对这个算法适当修改实现求累加和, 只需改动以下几点: (1) 使p 的初值为0 (为什么? 请读者考虑)。

(2) S3 中乘法运算改为加法运算。

(3) S5 中改为与100 比较。

由于计算机是高速进行运算的自动机器, 可以高速地、不厌其烦地进行重复性操作。

计算机中多次重复执行同类操作称为循环 (或迭代)。

所有计算机高级语言中都有实现循环的语句。

因此, 上述算法不仅是正确的, 而且是计算机能实现的较好的算法。

四、算法设计题例1.9 开发一个程序求全班学生某门课程的平均成绩。

要求程序每次运行时班级人数可为任意。

解 采用结构化程序设计方法 ( “自顶向下、逐步求精”的方法) 来解决这个问题并用伪代码来表示求精过程中得到的算法。

为使初学者比较容易读懂, 伪代码采用中文来表示。

首先, 顶层描述程序的整个功能, 然后将其分解为一系列较小的任务, 并按照执行顺序列出这些任务

。每一步的求精都是一个完整的算法说明, 只是每层的细节在变化。

重复上述求精过程直到表示每一任务的伪代码能够转化为程序设计语言的程序段为止。

对于上述问题, 顶层可用一句伪代码来表示: 求未知人数班级的平均成绩首先考虑需要使用的数据量 (变量)。

显然, 本例中需要使用的数据量有: 总成绩 (记为total) , 成绩个数计数器 (记为counter) , 每次输入的成绩 (记为score) , 全班平均成绩 (记为average) 。

对数据进行分析后, 再来考虑程序应该如何完成上述功能。

许多程序可从逻辑上分为三个阶段: 初始化阶段 (对程序变量初始化) ; 处理阶段 (输入数据值并对程序中的变量作相应调整) ; 终止阶段 (计算和打印最终结果) 。

本程序也不例外, 因此可以将顶层分解为以下三个小任务: 初始化变量; 输入、求和及对输入的成绩计数; 计算并打印班级平均成绩; 对初始化阶段, 由于每次输入的成绩值score 和平均成绩average 会被后面的读入过程覆盖, 故不需要初始化。

做如下求精: 初始化总成绩total为0; 初始化计数器counter为0; 由于本例中成绩个数是不定的, 可以使用一个专门的值来指示数据输入的结束, 这个值称为 “标记值” (flag) 。

显然标记值必须小心选择, 使它有别于准备接收的成绩值。

因为考试成绩通常是非负的, 可以选择-1 作为标记值。



## <<C语言程序设计导学>>

因此可以让程序重复读入成绩值，当遇到标记值时结束读入。  
每读入一个成绩值，就将其加到总成绩total中，计数器加1。  
&hellip;&hellip;

## <<C语言程序设计导学>>

### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>