



西京学院系列教材

# C语言程序设计

——实训教程

总主编 任万钧

主 编 魏燕明

副主编 张玉成 张 超



北京出版社



西京学院系列教材

ISBN 978-7-300-07633-5

# C语言程序设计

## ——实训教程

总主编 任万钧  
主 编 魏燕明  
副主编 张玉成 张 超

北京出版社

## 图书在版编目 (CIP) 数据

C 语言程序设计实训教程 / 魏燕明主编. —北京: 北京出版社, 2009. 1

ISBN 978 - 7 - 200 - 07633 - 2

I. C… II. 魏… III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 004764 号

### 内容简介:

本书是《C 语言程序设计——新编教程》的配套教材,主要包括实践篇和训练篇。在实践篇中主要分为上机实验和课程实训两部分。上机实验按照教材中章节的顺序安排了十三个具体实验,内容分别是:C 语言程序初步、数据类型、运算符与表达式、输入与输出、顺序和选择结构、条件型循环结构、计数型循环结构、数组、函数的定义及调用、指针、结构体和共用体、文件。课程实训是按照教育部的要求,为了提高学生的动手能力,为 C 语言课程周训所提供的 C 程序实训报告样式及学生成绩管理、奥运会倒计时、通讯簿管理、儿童加减乘除游戏的程序清单。在训练篇中主要提供了各章同步训练和教材的四套综合试卷及其相应的参考答案。

本教材在编写过程中强调实用性,作为学生课后复习、上机实验、课程实训的辅助教材,帮助学生掌握 C 语言的基本概念,提高阅读程序的能力,开阔编程思路和提高编程能力。该教材可作为大学本科应用型专业的学生以及高职高专学生学习“C 语言程序”课程的辅助教材,也可作为 C 语言自学者的参考书。

## C 语言程序设计——实训教程

C YUYAN CHENGXU SHEJI——SHIXUN JIAOCHENG

主编 魏燕明

\*

北京出版社出版

(北京北三环中路 6 号)

邮政编码:100120

网 址:www. bph. com. cn

北京出版社出版集团总发行

北京佳顺印务有限公司印刷

\*

787 × 1092 16 开本 9.5 印张 216 千字

2009 年 1 月第 1 版 2009 年 1 月第 1 次印刷

ISBN 978 - 7 - 200 - 07633 - 2

TP · 29 定价:13.00 元

质量监督电话:010 - 58572393

# 序

## 为全面提高教学质量助力

——为自编教材而作

当今世界，教育在综合国力的激烈竞争中处于非常重要的地位。国力强弱愈来愈取决于国家各类人才的质量和素质，这对于培养和造就适应我国经济社会发展需要的一大批专门人才和拔尖创新人才提出了更高的要求。

高等学校是高层次人才培养的重要阵地，是知识创新、技术创新的摇篮，是“科教兴国”的强大生力军。高校的根本任务是培养具有自主创新精神、创新能力、实践能力、德智体美全面发展的社会主义事业的合格建设者和可靠接班人。高等学校要完成好这一历史重任，就必须全面贯彻党的教育方针和落实科学发展观，努力提高教学质量，加强学科、专业建设和课程建设。其中一个重要任务之一是做好教材建设工作，编写出一大批规划教材和各种创新教材，确保教学质量，使高水平、高质量教材进课堂。这是因为教材是体现教学内容和教学方法的知识载体，是教学活动的基本工具，也是深化教育教学改革、全力推进素质教育、培养创新人才的重要保证。

为了实现这一任务，我们决心投入大量的人力、物力和财力。利用自己的专任教师，实行专兼结合，写出一批符合时代需要的新教材。我们所编教材要以科学发展观为指导，按照各类专业人才培养目标以及课程教学的基本要求和教学大纲的要求，结合实际，注重素质教育，有利于各种能力的培养。新教材要吸收国内外科学研究和教学研究的先进成果，将各学科的新知识、新理论和新技术充实到新教材中，关注新兴学科、交叉学科和新兴职业。一方面正确阐述各学科的基本理论、基本知识和基本技能，坚持理论联系实际，努力做到科学性、先进性、系统性、适用性和连续性的统一；另一方面符合教学规律和认知规律，富有启发性，有利于激发学生的学习兴趣，有利于学生知识、能力和素质的培养，有利于提高学生的学习能力、创新能力、实践能力、交流能力和社会适应能力。培养出既有远大理想和高尚职业道德，又有丰富科学技术、文化知识和敢于创新的高素质大学生奉献于时代，服务于社会，为“科教兴国”尽绵薄之力！

西京学院

任丁均 教授

## 前 言

我们已经知道，用 C 语言编写的过程程序就是一些数据定义和函数调用。要理解这种程序的含义，程序员必须掌握函数调用和函数实现的本身，这就是过程程序需要中间表示的原因。

本书的编写以培养学生 C 语言编程应用能力为主线，通过实验环节加强学生的实际动手能力。结合本学科发展的方向掌握流行的编程环境，为后续课程的学习打下坚实的基础。本书以指导实践教学为出发点，内容精炼。每个实验都有明确的实验目的、实验内容、实验习题等。

本书还为学生提供了每一章节的练习题和教程的综合试题，同时提供参考答案。用以全面巩固 C 语言所学的编程知识。

本教材由西京学院基础部魏燕明老师负责主编，计算机基础教研室全体教师参编。在本书的编写过程中得到了西安理工大学许淑贞老师的指导，在此表示衷心的感谢。同时，还参阅了大量的参考文献，在此对它们的作者一并表示感谢。

由于编者水平有限，加之时间仓促，书中错误之处在所难免，敬请读者批评指正。

编 者

2008 年 6 月 西安

## 目 录

上篇 实践篇 .....	1
第一部分 实验项目 .....	1
实验一 C 语言程序初步 .....	1
实验二 数据类型 .....	2
实验三 运算符与表达式 .....	3
实验四 输入与输出 .....	3
实验五 顺序与选择结构 .....	5
实验六 条件型循环结构 .....	6
实验七 计数型循环结构 .....	7
实验八 数组 .....	8
实验九 函数的定义及调用 .....	9
实验十 指针 (一) .....	10
实验十一 指针 (二) .....	12
实验十二 结构体和共同体 .....	12
实验十三 文件 .....	13
第二部分 C 语言常见错误分析与程序调试 .....	14
第一节 C 程序的常见错误 .....	14
一、变量、表达式和语句错误 .....	14
二、数组使用中的错误 .....	17
三、库函数调用错误 .....	18
四、自定义函数中出现的错误 .....	19
五、使用指针变量容易出现的错误 .....	21
六、其他常见错误 .....	22
第二节 C 程序的调试 .....	23
一、调试程序基本要领 .....	23
二、编译错误和连接错误的排除 .....	25
三、逻辑错误与运行错误的排除 .....	25
第三部分 C 语言课程实训 .....	28
第一节 系统需求分析 .....	28
第二节 系统设计 .....	28
一、各模块功能简介 .....	28
二、主界面设计 .....	29
第三节 设计代码 .....	29

一、版本信息界面设计代码 .....	29
二、菜单设计代码 .....	30
第四节 项目实训报告 .....	33
第五节 实训示例 .....	35
一、奥运会倒计时 .....	35
二、通讯簿管理系统 .....	35
小结 .....	39
附：实训题及其答案 .....	39
一、儿童加减乘除游戏项目 .....	39
二、显示万年历 .....	44

## 下篇 训练篇

第一部分 各章练习题 .....	50
第一章 练习题 .....	50
第二章 练习题 .....	53
第三章 练习题 .....	56
第四章 练习题 .....	60
第五章 练习题 .....	63
第六章 练习题 .....	69
第七章 练习题 .....	76
第八章 练习题 .....	85
第九章 练习题 .....	93
第十章 练习题 .....	99
第二部分 综合练习题 .....	102
综合试卷一 .....	102
综合试卷二 .....	107
综合试卷三 .....	111
综合试卷四 .....	114
第三部分 各章练习题答案 .....	118
第一章 练习题答案 .....	118
第二章 练习题答案 .....	118
第三章 练习题答案 .....	119
第四章 练习题答案 .....	120
第五章 练习题答案 .....	123
第六章 练习题答案 .....	124
第七章 练习题答案 .....	127
第八章 练习题答案 .....	130

---

第九章 练习题答案 .....	132
第十章 练习题答案 .....	135
<b>第四部分 综合练习题答案 .....</b>	<b>138</b>
综合试卷一答案 .....	138
综合试卷二答案 .....	140
综合试卷三答案 .....	141
综合试卷四答案 .....	142
<b>参考文献 .....</b>	<b>144</b>



# 上篇 实践篇

## 第一部分 实验项目

### 实验一 C 语言程序初步

#### 一、目的和要求

1. 熟悉 C 语言运行环境。
2. 掌握 C 语言程序的书写格式和 C 语言程序的结构。
3. 掌握 C 语言上机步骤，了解运行一个 C 程序的方法。
4. 本实验可在学习完教材第一章后进行。

#### 二、实验内容

##### 1. C 语言上机步骤

###### A. Turbo C2.0 环境

- (1) 启动 Turbo C: TC;
- (2) 编辑源文件;
- (3) 编译源文件: Compile/Compile to OBJ→Compile/Link EXE file→可执行文件 (.exe)以上两步也可合为一步: Compile/Make EXE file (或直接按功能键 F9);
- (4) 执行程序: File/Run (或直接按 Ctrl + F9);
- (5) 查看程序运行结果: Alt + F5。

###### B. VC++ 6.0 环境

- (1) 启动 VC;
- (2) 单击工具栏上的“新建文本文件”按钮;或者执行“文件/新建”命令,从弹出的“新建”对话框中选中“文件”选项卡,从列表框中选中“文本文件”,确定即可;
- (3) 输入源程序,保存。注意:①文件保存位置,最好不要放在桌面上②文件的扩展名一定是“.C”;
- (4) 单击工具栏上的“编译”按钮(^F7);或者执行“组建/编译”命令,在出现的对话框中单击“是”,于是在源程序的下面出现程序出错提示信息;
- (5) 修改程序,继续编译。程序通过后,单击工具栏上的“运行”按钮(^F5),即可出现运行结果。

2. 下面的 C 程序,是计算由键盘输入的任何两个整数的积。

```

/*****exl.c*****/
#include "stdio.h"

```

```

main ()
{
int x, y, z;
scanf ("%d%d", &x, &y);
z = x * y;
printf ("The product is :%d", z);
}

```

3. 练习教材中的程序。

### 三、实验步骤

1. 输入源程序，编译检查，如有错误，改正程序中的错误。
2. 编译并运行上述程序，记下所给出的出错信息。
3. 再编译执行纠错后的程序。如还有错误，再编辑改正，直到不出现语法错误为止。

### 四、分析与讨论

1. 记下在调试过程中所发现的错误、系统给出的出错信息和对策。分析讨论对策成功或失败的原因。
2. 总结 C 程序的结构和书写规则。

## 实验二 数据类型

### 一、目的和要求

1. 了解 C 语言中数据类型的意义。
2. 本实验可在学习了教材第二章后进行。

### 二、实验内容和步骤

1. 下面的程序试图计算由键盘输入的任意两个整数的平均值：

```

#include <stdio.h>
main ()
{
int x, y, a;
scanf ("%x,%y, &x, &y);
a = (x + y) /2;
printf ("The average is : " a);
}

```

分析上述程序有什么错误？若有错误，请指出错误原因。

调试到无语法错误后，分别使用下列几组数据对上述程序进行测试。

(1) 2, 6                      (2) -1, -2                      (3) 5.5, 6.9

2. 操作符 `sizeof` 用以测试一个数据或类型所占用的存储空间的字节数。请编写一个程

序, 测试各基本数据类型所占用的存储空间大小。

3. 练习新编教程教材中的相关例题与习题。

### 三、分析与讨论

如何正确地选用数据类型? (提示: 给合前面做过的两个实验及书本进行讨论总结)

## 实验三 运算符与表达式

### 一、目的和要求

1. 理解常用运算符的意义。
2. 掌握 C 语言表达式的运行规则。
3. 本实验可在学习了教材第三章后进行。

### 二、实验内容和步骤

1. 编写一个 C 语言程序, 测试下列各表达式的值

$i + 1, j + 1$

$i + +, j + +$

$+ + i, + + j$

要求在各表达式中  $i$  和  $j$  都分别有相同的初值。在实验中理解  $+$ ,  $++i$ ,  $i++$  的意义。

2. 编写一个程序, 测试常用的算术运算符的优先顺序。
3. 练习新编教程教材中的相关例题与习题。

### 三、分析与讨论

分析总结运算符的优先级。

## 实验四 输入与输出

### 一、目的和要求

1. 掌握 C 语言程序输入、输出的方法和格式。
2. 本实验可在学习教材第二、三章后进行。

### 二、实验内容与步骤

1. 输入并编辑下面的程序。这个程序有语法错误吗? 为什么? 请改正。

```
main ()
{
int a, b;
float c, d;
```

```

long e, f;
unsigned int u, v;
char c1, c2;
scanf ("%d,%d", a, b);
scanf ("%f,%f", c, d);
scanf ("%ld,%ld", e, f);
scanf ("%o,%o", u, v);
scanf ("%c,%c", c1, c2);
printf ("\n");
printf ("a=%7d, b=%7d\n", &a, &b);
printf ("c=%10.2f, d=%10.2f\n", &c, &d);
printf ("e=17ld, f=%17ld\n", &e, &f);
printf ("u=%o, d=%o\n", &u, &v);
printf ("c1=%c, d=%c\n", &c1, &c2);

```

调试上述程序无语法错误后,用下面的测试数据,对程序进行测试:

a = 123, b = 456, c = 17.6, d = 71837.65, e = 70000, f = 2174506, u = 62000,  
v = 58765, c1 = 'a', c2 = 'b'

分析运行结果。特别注意输入 c1, c2 的值是什么? 什么原因?

2. 将输入 e 和 f 的语句改为:

```
scanf ("%d,%d", &e, &f);
```

再用上述测试数据测试并分析结果。

3. 将输入 u、v 的语句改为:

```
scanf ("%d,%d", &u, &v);
```

再用上述测试数据测试并分析结果。

4. 将输出 e, f 的语句改为:

```
printf ("e=%17d, f=%17d\n", e, f);
```

再用上述测试数据测试并分析结果。

5. 将输出 u、v 的语句改为:

```
printf ("u=%u, v=%u\n", u, v);
```

或

```
printf ("u=%d, v=%d\n", u, v);
```

再用上述测试数据测试并分析结果。

6. 请读者自己修改程序和改变数据输入的形式,分析各种情况下的输入与输出。

### 三、分析与讨论

1. 总结在 printf 函数中可以使用的各种格式指定符,并给出例子。

2. 总结在 printf 函数中可以使用的各转义字符及其功能。例如转义字符 \n 的意义?

## 实验五 顺序和选择结构

### 一、实验目的

1. 了解条件与程序流程的关系。
2. 了解用不同的数据使程序的流程覆盖不同的语句、分支和路径。
3. 本实验应在学习教材第四章后进行。

### 二、实验内容和步骤

1. 编写输入三角形的三边长  $a$ 、 $b$ 、 $c$ ，求三角形面积  $area$  的程序。

要求：

- (1) 输入前要加提示语句。
- (2) 输出结果前要有必要的文字说明。
- (3) 输入一组数据 3 4 5，观察运算结果。
- (4) 输入另外一组数据 3 4 8，观察运算结果，分析这个运算结果是否有效。

```
#include <stdio.h>
#include "math.h"
main ()
{float a, b, c, p, area;
printf (" \n 请输入三角形的三边长:");
scanf ("%f%f%f", &a, &b, &c);
if (a + b > c && a + c > b && b + c > a)
{p = (a + b + c) / 2;
area = sqrt (p * (p - a) * (p - b) * (p - c));
printf (" \n 三角形的面积为: S = %. 2f", area);}
else
printf (" \n 不能构成三角形!");
}
```

### 三、分析讨论

(1) 用哪一组数据就可使程序中的每个处理语句都执行一次？为了找出程序中各条处理语句中的错误，应该使用什么样的数据对程序进行测试？请上机验证自己的结论。

(2) 用哪两组数据就可以使程序段中的每个分支都运行一次？这种情形与上面的讨论有何不同？如果为了找出程序中积压分支中的错误，应该使用什么样的数据对程序进行测试？请上机验证自己的结论。

### 四、进一步的实验

1. 题目：从键盘上输入三个数，让它们代表三条线段的长度，请写一个判断这三条线段所组成的三角形属于什么类型（不等边，等腰，等边或不构成三角形）的 C 程序。

2. 编辑一个分支控制程序，求：任意输入的数 ( $n > 2$ ) 是否为素数。
3. 编辑一个分支控制程序，求：任意输入的 3 个数中的最大数。
4. 教材中的程序。

## 实验六 条件型循环结构

### 一、目的和要求

1. 掌握在程序设计条件型循环结构时，如何正确地设定循环条件，以及如何控制循环的次数。
2. 了解条件型循环结构的基本测试方法。
3. 本实验可在学习教材第五章后进行。

### 二、实验内容与步骤

1. 下面是一个计算  $e$  的近似值（使误差小于给定的  $\delta$ ）的程序。

```
#include <stdio.h>
main ()
{ double e = 1.0, x = 1.0, y, detax;
  int i = 1;
  printf ( " \ n please input enter a error: " );
  scanf ( " %lf" , &detax);
  y = 1/x;
  while ( y >= detax)
  { x = x * i;
    y = 1/x;
    e = e + y;
    ++i; }
  printf ( " %12.10lf" , e);
}
```

2. (1) 阅读上面的程序，写出程序所依据的计算公式。  
(2) 当输入的  $detax$  各是什么值时，能分别使程序按下面的要求运行：  
不进入循环；只循环一次；只循环两次；进入死循环（程序将永远循环下去）。  
为了能知道程序循环了多少次，应该在程序中增加一条什么样的语句？  
(3) 源程序中  $while$  语句中的  $y \geq detax$ ，分别换成  $y > detax$ ， $y = detax$ ， $y < detax$ ， $y \leq detax$ ，观察程序运行将会有有什么变化。  
(4) 把原程序中  $while$  语句之前的  $y = 1/x$  语句去掉，观察程序的运行将会发生什么样的变化。假如不知道机器内的程序实际上是怎么写的，输入什么样的  $detax$  就能测试出少了上述这条语句。  
(5) 若把原程序中的  $++i$  换成  $i++$ ，观察程序的运行发生了什么变化？  
(6) 把原程序中的  $while$  结构改写成  $do-while$  结构，再分别按上述的 (2)、(3) 两

步进行实验。

3. 编辑一个循环程序，求：1 到 100 的和及平方和。
4. 练习新编教程教材中的相关例题与习题。

### 三、分析讨论

总结一下测试条件循环结构的一般方法。

## 实验七 计数型循环结构

### 一、目的和要求

1. 掌握如何正确地控制计数型循环结构的次数。
2. 了解对计数型循环结构进行测试的基本方法。
3. 了解在嵌套循环结构中，提高程序效率的方法。
4. 本实验应在学习完教材第三、四章后进行。

### 二、实验内容

1. 计算一个正整数 num 的各位数字之和。

```
#include "stdio. h"
void main ()
{ long n, k = 0;
printf ( " \ nplease input a number: " );
scanf ( "%ld" , &n);
do
{ k = k + ( n%10);
n = n/10;} while ( n);
printf ( " \ n%ld \ n" , k);}
```

2. 计算  $1 + 2 + \dots + n$ 。

```
#include "stdio. h"
void main ()
{ int i, sum, n;
scanf ( "%d" , &n);
for ( i = 1; i <= n; i++)
sum = sum + i;
printf ( " \ nsum = %d \ n" , sum);}
```

3. 将输入的一串字符显示在屏幕上，遇到字符 "\$" 时结束。

```
#include <stdio. h >
void main ()
{ char c;
printf ( "Please input a string: " );
```

```

for (c = getchar (); c != '$'; c = getchar ())
    putchar (c);}
4. 计算  $s = 1! + 2! + \dots + n!$ 
#include <stdio. h >
void main ()
{int i = 1, p = 1, s = 0, n; /* 定义变量, 并赋初值 */
scanf ("%d", &n);}
for (; i <= n;)
{p * = i;          /* 求 n! */
s + = p;          /* 求 1! + 2! + 3! + ... + n! */
i ++;}
printf ("\ nsum = %d \ n", s);}

```

5. 编写程序在一行内输出整数 1 到 10, 并连续输出 5 行。

```

#include <stdio. h >
void main ()
{int i, j;
for (i = 1; i <= 5; i ++ )
{for (j = 1; j <= 10; j ++ )
printf ("%d ", j);
printf ("\ n");}}

```

6. 练习新编教程教材中的相关例题与习题。

### 三、分析讨论

1. 如何测试计数型循环结构的控制表达式中的错误?
2. 从实验中你得到了哪些提高嵌套循环程序效率的启示?

## 实验八 数组

### 一、目的和要求

1. 掌握数组定义的规则。
2. 掌握 C 语言数组的基本用法。
3. 本实验可在学习教材第六章后进行。

### 二、实验内容与步骤

1. 编写程序, 测试下列数组的定义方式是否正确。

```

(1) main ()
{int n;
scanf ("%d", &n);
int a [n];

```



```
.....}
```

```
(2) #include <stdio. h >
```

```
#define M 10
```

```
main ( )
```

```
{ int a [ M ];
```

```
.....}
```

通过这一实验，可以说明什么问题？

2. 运行下面的 C 程序，根据运行结果，可以说明什么？

```
main ( )
```

```
{ int num [5] = {1, 2, 3, 4, 5};
```

```
int i;
```

```
for (i=0; i<5; i++)
```

```
printf ("%d", num [i]);
```

```
}
```

3. 操作符 & 用以求一个变量的地址，这在函数 scanf 中已经使用过了。现在要你设计一个程序，返回一个 3 \* 5 的二维数组各元素的地址，并由此说明二维数组中各元素是按什么顺序存储的。

4. 任意输入 20 个数在一维数组 a 中，求 20 个数的平均值。

5. 定义 2 个二维数组 A [2] [3]、B [2] [3]，分别对这 2 个数组输入数据，求 A + B 和 A - B 的值。

6. 用字符数组存储任意输入的 10 字符，求其最大字符。

7. 编辑一个程序：有 5 个学生，考 3 门功课。求出每一个学生的总分及平均分，统计平均成绩低于 60 分的学生人数。

8. 为一个起泡排序程序设计测试用例，并测试之。例如：已知一组数据如下：6, 3, 42, 23, 35, 71, 98, 67, 56, 38。编写程序，把它们按从小到大的次序排列起来。

9. 求 2 行 3 列矩阵 A 的转置矩阵。

10. 练习新编教程教材中的相关例题与习题。

### 三、分析讨论

通过实验，分析定义与引用数组的区别。

## 实验九 函数的定义及调用

### 一、目的与要求

1. 掌握 C 语言函数定义及调用的规则。
2. 理解参数传递的过程。
3. 本实验可在学习了教材第七章后进行。

### 二、实验内容与步骤

1. 上机调试下面的程序。