

C 语言 程 序 设 计 系 列

吕凤翥 张静波 编著

C 语言程序设计

习题解答与上机指导

<http://www.tup.com.cn>

清华大学出版社



C 语 言 程 序 设 计 系 列

吕凤翥 张静波 编著

C 语言程序设计

习题解答与上机指导

清华大学出版社
北京

内 容 简 介

本书是《C 语言程序设计系列》丛书的一部分。它与该系列中的《C 语言程序设计——基础理论与案例》(简称“教材”)一书配合使用。本书归纳了教材中各章的主要内容,给出了每章练习题的参考答案,又为每一章编写了上机指导,以帮助读者上机练习。本书每章还备有测试题。

本书适合作为高等院校理科学生学习 C 语言课程的教材,也可作为自学 C 语言读者的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C 语言程序设计习题解答与上机指导/吕凤翥,张静波编著. —北京:清华大学出版社,2006.2
(C 语言程序设计系列)

ISBN 7-302-12439-6

I. C… II. ①吕… ②张… III. C 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 005679 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
http://www.tup.com.cn 邮 编:100084
社 总 机:010-62770175 客 户 服 务:010-62776969

组稿编辑:丁 岭

文稿编辑:孙建春

印 装 者:北京嘉实印刷有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印 张:10 字 数:244 千字

版 次:2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷

书 号:ISBN 7-302-12439-6/TP·7978

印 数:1~3000

定 价:18.00 元

前言

本书是《C 语言程序设计系列》丛书的一部分。它与该系列中的《C 语言程序设计——基础理论与案例》(简称“教材”)一书配合使用。本书归纳了教材中每一章的主要内容,给出了每章练习题的参考答案,又为每一章编写了上机指导,帮助读者上机练习,每章还备有测试题。

做练习题是学好计算机语言课的必要环节。该系列中《C 语言程序设计——基础理论与案例》一书中每章配有丰富的练习题,供读者选用。通过做这些练习题来检验学过的知识,巩固讲过的内容,帮助读者进一步复习和提出还没有学懂的问题。练习题都是宝,通过练习才能把学过的知识记得牢。为了检查做过的习题是否做得对,本书提供了所有练习题的参考答案。对照答案可发现做错问题的原因,进一步复习或请教,把出现的错误改正,有助于对学过知识的巩固和提高。通过做练习题,可以巩固学过的概念,记住语法规则,特别是通过分析程序结果和编程的练习,学会编程技巧,积累编程方法,为今后的编程打下基础。

上机调试是学好计算机语言课的重要方法。许多同学问,怎样才能学好计算机语言课,我经常是这样回答:边看书,边上机。有些计算机语言我就是这样学会的。听课、看书是很重要的,上机调试也很重要。通过上机可以验证某些概念,可以学会语法规则,可以检验编程能力。有些想不明白的问题可以上机试试,实践会告诉我们答案。例如,数组初始化时是否判越界?要想知道答案,可以先上机试一下,对一个数组进行初始化,使用的初始化表中数据项的项数超过该数组元素个数时,会发生什么现象?在实际学习过程中会发现许多一时搞不清的问题,有些问题在书中又不能直接找到答案,这时怎么办?那么,请上机试试。例如,运算符增 1 和减 1 只能作用在变量上,不能作用在常量和表达式上。真的不能作用在常量和表达式上吗?上机一试便知道:使用++或--运算符作用在常量和表达式上是会出现错误的。这样通过实践就会牢记这一概念。本书为了帮助读者上机实践,每章都提出了上机要求,并编排了上机练习题,若读者能够按照要求上机练习,一定会提高编程能力和水平。而每章练习题中的分析程序结果题和编程题,都特别要求读者上机调试。

为了检验读者学习 C 语言程序设计的实际效果,本书每章又提出一些练习题供选做。

读者读了本书后,还有什么不满意的地方,或发现了错误,均请予指教。作者表示十分感谢。

作者

2005 年夏写于北大燕北园

目 录

第1章 C语言概述	1
1.1 本章主要内容	1
1.1.1 C语言的由来和发展	1
1.1.2 C语言的特点和应用	1
1.1.3 C语言程序的特点	2
1.1.4 标准文件的读写函数	2
1.1.5 C语言程序的实现	3
1.2 练习题1 参考答案	4
1.3 测试题	8
1.4 上机练习和指导	10
1.4.1 本章上机要求	10
1.4.2 上机练习题	10
第2章 单词、常量和变量	12
2.1 本章主要内容	12
2.1.1 单词	12
2.1.2 类型	12
2.1.3 常量	13
2.1.4 变量	14
2.2 练习题2 参考答案	14
2.3 测试题	19
2.4 上机练习和指导	21
2.4.1 本章上机要求	21
2.4.2 上机练习题	21
第3章 运算符和表达式	22
3.1 本章主要内容	22
3.1.1 运算符的功能	22
3.1.2 运算符的优先级和结合性	24
3.1.3 表达式的值和类型	25
3.1.4 表达式求值时应注意的问题	25
3.1.5 数据类型转换	26
3.2 练习题3 参考答案	26

3.3	测试题	30
3.4	上机练习和指导	31
3.4.1	本章上机要求	31
3.4.2	上机练习题	31
第 4 章	预处理功能	33
4.1	本章主要内容	33
4.1.1	预处理命令的特点	33
4.1.2	宏定义命令	33
4.1.3	文件包含命令和条件编译命令	34
4.1.4	类型定义	35
4.2	练习题 4 参考答案	36
4.3	测试题	42
4.4	上机练习和指导	44
4.4.1	本章上机要求	44
4.4.2	上机练习题	44
第 5 章	语句	46
5.1	本章主要内容	46
5.1.1	C 语言语句概述	46
5.1.2	分支语句	46
5.1.3	循环语句	47
5.1.4	转向语句	48
5.1.5	程序举例	49
5.2	练习题 5 参考答案	49
5.3	测试题	59
5.4	上机练习和指导	62
5.4.1	本章上机要求	62
5.4.2	上机练习题	62
第 6 章	数组	64
6.1	本章主要内容	64
6.1.1	数组的定义格式和数组元素表示方法	64
6.1.2	数组的赋值和运算	65
6.1.3	字符数组	66
6.1.4	数组应用举例	67
6.2	练习题 6 参考答案	67
6.3	测试题	74
6.4	上机练习和指导	75

6.4.1 本章上机要求	75
6.4.2 上机练习题	75
第 7 章 指针	77
7.1 本章主要内容	77
7.1.1 指针及其定义格式	77
7.1.2 指针的赋值和运算	77
7.1.3 数组元素的指针表示	78
7.1.4 指向数组的指针和指针数组	79
7.1.5 字符指针	80
7.2 练习题 7 参考答案	80
7.3 测试题	89
7.4 上机练习和指导	91
7.4.1 本章上机要求	91
7.4.2 上机练习题	92
第 8 章 函数	93
8.1 本章主要内容	93
8.1.1 函数的定义和说明	93
8.1.2 函数的参数和返回值	93
8.1.3 函数的调用	94
8.1.4 指针作函数参数和返回值	95
8.1.5 系统函数	95
8.2 练习题 8 参考答案	96
8.3 测试题	109
8.4 上机练习和指导	111
8.4.1 本章上机要求	111
8.4.2 上机练习题	112
第 9 章 结构和联合	113
9.1 本章主要内容	113
9.1.1 结构和结构变量的定义格式	113
9.1.2 结构变量的赋值	113
9.1.3 结构数组	114
9.1.4 结构变量和指向结构变量指针的应用	114
9.1.5 结构和链表	114
9.1.6 联合	115
9.1.7 枚举	115
9.2 练习题 9 参考答案	116

9.3	测试题	127
9.4	上机练习和指导	130
9.4.1	本章上机要求	130
9.4.2	上机练习题	130
第 10 章	文件操作	132
10.1	本章主要内容	132
10.1.1	C 语言文件操作的特点	132
10.1.2	标准文件的读写操作	132
10.1.3	一般文件的操作	133
10.1.4	文件检测函数	134
10.2	练习题 10 参考答案	135
10.3	测试题	146
10.4	上机练习和指导	148
10.4.1	本章上机要求	148
10.4.2	上机练习题	148

第 1 章 C 语言概述

1.1 本章主要内容

1.1.1 C 语言的由来和发展

1. C 语言的由来

C 语言由来的重要知识点如下：

- 诞生于 1972 年，于美国贝尔实验室。
- 在 B 语言基础上改进的。

2. C 语言的发展

C 语言的发展经历了以下过程：

- 1973 年使用 C 语言修改了 UNIX 操作系统。
- 1978 年 C 语言被移植到大、中、小和微型计算机上。
- 1987 年公布了 87 ANSI C 新版本。
- 1983 年在 C 语言基础上推出 C++ 语言。

1.1.2 C 语言的特点和应用

1. C 语言的特点

C 语言有以下特点：

- C 语言是一种结构化的程序设计语言。
- C 语言程序简洁明了、方便灵活。
- C 语言功能强大，具有高级语言和低级语言的特征。
- C 语言可移植性好。
- C 语言也有一些不足。

2. C 语言的应用

C 语言的应用包括以下几个方面：

- 数据库管理。
- 图形图像系统。
- 编写设备接口。

- 数据结构。
 - 排序和检索。
- 总之，C 语言应用广泛。

1.1.3 C 语言程序的特点

1. C 语言程序举例

《C 语言程序设计——基础理论与案例》(后简称“教材”)中与此内容相对应的例子为：

【例 1.1】和【例 1.2】。

2. C 语言程序在结构上的特点

C 语言程序在结构上有如下特点：

- 一个 C 语言程序是由一个或多个文件组成的，每个文件是由一个或多个函数组成的，函数是构成 C 语言程序的最小模块。
- C 语言程序实际上是一个函数串。在组成 C 语言程序的若干个函数中，有且仅有一个是主函数。
- C 语言程序是从主函数开始执行的，程序中的其他函数都是被主函数或是由主函数调用的函数调用的。函数之间是调用关系，不能在一个函数中定义另一个函数。
- 函数是由函数头和函数体构成的，函数包括函数类型、函数名、圆括号和参数表；函数体是由一对花括号括起的若干条语句组成的。
- 语句是由分隔符进行分隔的若干个单词组成的。

3. C 语言程序在书写上的约定

由于 C 语言程序可读性较差，要求按下列习惯书写 C 语言程序。

- 通常一行写一条语句，有些语句可写多行。
- 规定每条语句以分号结束。
- 花括号按规定格式书写。
- 采用缩进方法。
- 适当使用注释信息。

1.1.4 标准文件的读写函数

1. 标准文件的读函数

标准文件的读函数包括：

- `getchar()`
- `gets()`
- `scanf()`

`scanf()`函数格式如下：

```
int scanf ("<控制串>", <参数表>)
```

教材中解释了如下内容。

- <控制串>中格式符的规定和一般字符的用途。
- <参数表>中对参数的要求, 以及<参数表>中参数的个数和类型与<控制串>中格式符的关系。

2. 标准文件的写函数

标准文件的写函数包括:

- putchar ()
- puts ()
- printf ()

printf ()函数格式如下:

```
int printf ("<控制串>", <参数表>)
```

教材中解释了如下内容。

- <控制串>中格式符的规定和一般字符的用途。
- <参数表>中对参数的要求, 以及<参数表>中参数的个数和类型与<控制串>中格式符的关系。

思考题:

- (1) 如何在键盘的输入数据流中获取一个双精度浮点数给一个变量赋值?
- (2) 如何将一个字符串常量输出显示到屏幕的当前光标处?
- (3) 如何在屏幕的当前光标处输出显示一个字符?

1.1.5 C语言程序的实现

1. C语言程序实现的三步曲

C语言程序实现的三步曲包括:

- 编辑。
- 编译连接。
- 运行。

2. Microsoft Visual C++ 6.0 系统用来实现 C/C++语言程序的功能

主要功能有:

- 编辑源文件。
- 存储源文件。
- 编译连接源文件。
- 运行可执行文件。
- 创建空的项目文件。

- 将多个文件添加到项目文件中。
- 编译连接和运行项目文件。

3. C 语言程序的具体实现

C 语言程序的具体实现包括:

- 单文件程序的实现。
- 多文件程序的实现。

思考题:

- (1) 带有“#”号的预处理命令在何时执行?
- (2) 在向空项目文件中添加多个文件时, 添加文件的先后顺序是否有关?

本章作业: 练习题 1 中一~六题。

1.2 练习题 1 参考答案

一、判断下列描述是否正确, 对者划√, 错者划×。

1. C 语言是一种非结构化程序设计语言。()
2. C 语言是一种面向对象的程序设计语言。()
3. C 语言具有简洁明了的特点。()
4. C 语言是一种具有某些低级语言特征的高级语言。()
5. C 语言数据类型转换比较灵活, 类型的使用不够严格。()
6. C 语言的应用比较广泛, 不仅可用来编写应用软件, 而且可用来编写系统软件。()
7. C 语言程序只能有一个文件。()
8. C 语言程序中可以有多个函数, 但只能有一个主函数。()
9. C 语言程序在书写上只能是一条语句写一行。()
10. C 语言程序的可读性较差, 因此要求书写 C 语言程序应按照习惯上的约定。()
11. C 语言标准输入操作中, gets()函数一次只可读取一个字符。()
12. C 语言标准输出操作中, putchar()函数可以输出显示一个字符串。()
13. C 语言标准格式输入函数 scanf()的参数表中要使用变量的地址值。()
14. C 语言标准格式输出函数 printf()的参数表是由若干个表达式组成。()
15. C 语言程序实现与其他高级语言一样也要经过编辑、编译连接和运行这样的三部曲。()

【答】对者有 3, 4, 5, 6, 8, 10, 13, 14, 15。

二、单选题。

1. 下列对 C 语言特点的描述中, 错误的是 ()。
A. C 语言不是结构化程序设计语言 B. C 语言编程简洁明了
C. C 语言功能较强 D. C 语言移植性好

【答】A

2. 下列关于C语言程序在结构上特点的描述中, 错误的是()。

- A. C语言程序是一个函数串
- B. C语言程序可以由一个或多个文件组成
- C. C语言程序的诸多函数中, 只能有一个主函数
- D. C语言程序被执行时, 通常是从第1个函数开始

【答】D

3. 在scanf()函数的格式说明符中, 长整型数的输入格式说明符是()。

- A. %d
- B. %lf
- C. %ld
- D. %x

【答】C

4. 在printf()函数的格式说明符中, 双精度浮点型数的输出格式说明符是()。

- A. %ld
- B. %lf
- C. %c
- D. %s

【答】B

5. 将一个双精度浮点型常数12.345使用printf()函数输出显示在屏幕上, 要求输出宽度为数值左边有2个空格符, 小数点后边有4位数, 正确的输出语句是()。

- A. printf("%9.4lf\n", 12.345);
- B. printf("%8.4lf\n", 12.345);
- C. printf("%lf\n", 12.345);
- D. printf("%9.4f\n", 12.345);

【答】A

三、分析下列程序的输出结果。

1.

```
#include <stdio.h>
void main()
{
    printf("I love the Beijing!\n");
    printf("We are students.\n");
}
```

【答】I love the Beijing!

We are students.

2.

```
#include <stdio.h>
void main()
{
    int a;
    a=5;
    printf("a=%d\n", a);
}
```

【答】a=5

3.

```
#include <stdio.h>
void main()
{
    char s[50];
    printf("Enter a string: ");
    gets(s);
    puts(s);
}
Enter a string: string ✓
```

【答】 string

4.

```
#include <stdio.h>
void main()
{
    int a;
    double d;
    printf("Enter a and d: ");
    scanf("%d%lf", &a, &d);
    printf("%5d,%8.2lf\n", a, d);
}
Enter a and d: 56 15.765 ✓
```

【答】 56, 15.77

5.

```
#include <stdio.h>
void main()
{
    char c='p';
    printf("%d,%c\n", c, c);
    printf("%d,%c\n", 'a', 'a');
}
```

【答】 121, p
97, a

四、编程题。

编程输出下列图案:

```
* * * * *
?   M E N U   ?
$ $ $ $ $ $ $ $
```

【解】 程序内容如下:

```
#include <stdio.h>
void main()
{
    printf ("*****\n");
    printf ("?  MENU  ? \n");
    printf ("$$$$$$\n");
}
```

五、修改下列程序中的错误，上机调试得到正确结果。

```
#include <stdio.h>
void main()
{
    int a=10;
    gets(s);
    print(s,s);
    print(f,a);
}
```

【解】修改后的程序如下：

```
#include <stdio.h>
void main ()
{
    int a=10;
    char s[50];
    printf ("Enter a string :");
    gets (s);
    printf ("%s\n",s);
    printf ("%d\n",a);
}
```

六、简答题。

1. C语言有哪些特点？

【答】C语言的特点如下：

- (1) C语言是一个结构化程序设计语言。
- (2) C语言编程简洁明了。
- (3) C语言功能很强。
- (4) C语言移植性很好。
- (5) C语言也有一些不足。

2. 为什么说C语言是一种不完全的结构化程序设计语言？

【答】从下述两点可看出C语言是一种不完全的结构化程序设计语言：

- (1) 可以使用 `goto` 语言在一个函数体内。
- (2) 允许一个函数中使用多个 `return` 返回语句。

3. C 语言程序结构上有何特点?

【答】C 语言程序在结构上有如下特点:

- (1) C 语言程序是由一个或多个文件组成, 每个文件是由一个或多个函数组成。
- (2) C 语言程序是一个函数串, 在诸多的函数中, 有且仅有一个主函数 `main()`。
- (3) C 语言程序中的函数之间关系是调用关系, 任何一个函数都将被主函数或主函数所调用的函数所调用。
- (4) 函数是 C 语言程序中最小的结构化模块。函数是由函数头和函数体组成。函数头中有类型说明、函数名和函数参数表; 函数体是由若干条语句组成的。

4. C 语言中, 标准输出设备是什么? 标准设备输出操作有哪些常用函数?

【答】C 语言中, 标准输出设备是显示器的屏幕。标准设备输出操作的函数有如下 3 个:

- (1) 输出一个字符的函数 `putchar()`。
- (2) 输出一个字符串的函数 `puts()`。
- (3) 格式输出函数 `printf()`, 它可以按格式输出各种不同数据类型的表达式的值。

5. 如何实现单文件的 C 语言程序? 如何实现多文件的 C 语言程序?

【答】实现单文件 C 语言程序的方法如下:

- (1) 编辑该文件, 并存储于磁盘。存盘时扩展名用 `.C`。
- (2) 使用 Build 菜单中的 Build 命令, 进行编译连接, 有错修改, 直到无错, 生成可执行文件。
- (3) 使用 Build 菜单中的 Execute 命令, 或者工具栏中相关按钮, 运行可执行程序。运行结果被显示在 DOS 屏幕上。

实现多文件 C 语言程序的方法如下:

- (1) 单独编辑每个文件, 并存于磁盘。
- (2) 使用 New 对话框中 project 选项卡, 建立一个空的项目文件。
- (3) 将已编辑好的文件添加到项目文件中。使用主窗口菜单栏中 project 菜单中的 Add To Project 选项, 单击该选项的 Files 命令, 在 Insert File into Project 对话框中将已编好的若干个 C 语言源文件加入到空的项目文件中。
- (4) 对项目文件进行编译连接, 然后运行。其方法同单文件程序操作。

1.3 测 试 题

一、判断下列描述是否正确, 对者划√, 错者划×。

1. C 语言具有地址运算、位操作等特点。()
2. 标准格式输入函数 `scanf()` 可以从键盘上接收不同数据类型的数据项。()

3. C语言的书写格式不会影响可读性。()
4. 使用 Visual C++ 6.0 版本编译系统不能够编译 C 语言的源程序。()
5. 使用 Visual C++ 6.0 版本编译系统编译 C 语言程序时, 如果发现程序中有警告错, 则不能生成可执行文件。()

二、单选题。

1. 下列关于 printf() 函数的描述中, 错误的是 ()。
 - A. 该函数是标准格式输出函数
 - B. 该函数参数表是由若干个表达式组成的
 - C. 该函数控制串中的格式符的个数应与参数表中参数的个数相等
 - D. 该函数参数表中各个参数的计算顺序一定是从左至右的
2. 下列转义字符中, 退格符是 ()。
 - A. \t
 - B. \r
 - C. \b
 - D. \a
3. 下列是输出显示字符'a'的八进制 ASCII 码的语句, 正确的是 ()。
 - A. printf("%c\n", 'a');
 - B. printf("%d\n", 'a');
 - C. printf("%s\n", 'a');
 - D. printf("%o\n", 'a');

三、填空题。

1. 一个 C 语言程序是由若干个_____组成的, 每个文件是由若干个_____组成的。_____函数只能有一个。
2. C 语言程序中, 每条语句必须以_____结束。主函数的位置可以_____。
3. 已知: int a=1; printf("%n", 'a'); 则输出结果是_____。

四、分析下列程序的输出结果。

1.

```
#include <stdio.h>
void main ()
{
    float a=3.12345;
    printf ("%2f,%2e\n", a, a);
    printf ("%2g\n", a);
}
```

2.

```
#include <stdio.h>
void main ()
{
    int i;
    printf ("Enter i: ");
    scanf ("% x", &i );
```