

C YUYAN CHENGXU SHEJI SHANGJI ZHIDAO



主 编 吴国凤
副主编 宣善立

C 语言程序设计

上机指导

YUYAN
CHENGXU SHEJI
SHANGJI ZHIDAO

中国科学技术大学出版社

C 语言程序设计上机指导

主 编：吴国凤

副主编：宣善立

主 审：胡学钢 王 浩

中国科学技术大学出版社

2003 • 合肥

图书在版编目 (CIP) 数据

C 语言程序设计上机指导/吴国凤主编. —合肥: 中国科学技术大学出版社, 2003.2
ISBN 7-312-01148-9

I. C… II. 吴… 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 (2003) 第 009884 号

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路 96 号, 邮编: 230026)

中国科学技术大学印刷厂印刷

全国新华书店经销

开本: 787×1092/16 印张: 7.5 字数: 190 千

2003 年 2 月第 1 版 2003 年 2 月第 1 次印刷

印数: 1—5 000 册

ISBN 7-312-01148-9/TP·244 定价: 10.00 元

前 言

掌握计算机知识和应用，是培养新型人才的一个重要环节。学习计算机知识不仅是为了掌握一种技能，更重要的是启发人们对先进科技的向往，激发创新意识，推动对新知识的学习，培养自学能力，锻炼动手实践的本领，这是高等学校全面素质教育中极为重要的一部分。

为了更好的帮助学习者掌握程序设计的方法和编写程序的技巧，提高独立思考问题和解决问题的能力为此编写了《C 语言程序设计上机指导》，此书是《C 语言程序设计教程》的配套用书。

C 语言程序设计是一门实践性很强的计算机语言，在学习过程中，检查学习好坏的标准，不是“知道不知道”，而是“会不会用”，学习的目的全在于应用。因此，希望学习者一定要重视实践环节，多上机练习，千万不要满足于“上课能听懂，教材能看懂”。有一些问题，别人说半天也不明白，自己一上机就清楚了。因而一定要多上机，从中进一步提高 C 语言程序设计能力。

全书各章主要包括两个方面的内容：一是对各章的知识要点作了详细说明；二是为各章配备了不同的实验。全书共安排了 21 个实验，每个实验都给出了实验目的、实验要求和实验内容。

本书由吴国凤、宣善立、谢文佩、王金玲编写。全书由吴国凤、宣善立统编定稿，由胡学钢、王浩主审。由于时间比较仓促，书中难免有疏忽、错误之处，恳请读者批评指正。

本书在编写过程中，得到兄弟高校计算机基础教育教师的关心和帮助，教研室的同仁们提出了许多宝贵意见；专家顾问们给予了悉心指导；在出版过程中，得到了中国科学技术大学出版社和合肥工业大学教材科的极大帮助，在此一并表示衷心的感谢。

编 者

2003 年 1 月

目 录

前言	(I)
第一章 C 语言概述	(1)
本章要点	(1)
实验 1 C 程序的运行环境和编辑一个 C 程序的方法	(1)
第二章 数据类型与运算规则	(4)
本章要点	(4)
实验 2 数据类型、运算符和表达式	(5)
第三章 程序控制结构	(9)
本章要点	(9)
实验 3 顺序程序设计	(11)
实验 4 分支结构程序设计	(13)
实验 5 循环结构程序设计	(16)
实验 6 程序设计综合应用	(18)
第四章 数组	(21)
本章要点	(21)
实验 7 一维数组的应用	(22)
实验 8 二维数组的应用	(25)
实验 9 字符数组的应用	(27)
实验 10 数组综合应用	(29)
第五章 函数	(32)
本章要点	(32)
实验 11 函数的传递方式	(34)
实验 12 变量的存储类型	(36)
实验 13 函数的嵌套调用和递归调用	(39)
实验 14 编译预处理	(41)
实验 15 函数综合应用	(43)
第六章 指针	(46)
本章要点	(46)

实验 16 指针应用一	(48)
实验 17 指针应用二	(52)
第七章 结构体与联合	(55)
本章要点	(55)
实验 18 结构体与联合应用一	(56)
实验 19 结构体与联合应用二	(58)
第八章 位运算	(61)
本章要点	(61)
实验 20 位运算应用	(61)
第九章 文件	(63)
本章要点	(63)
实验 21 文件的应用	(63)
实验报告	(67)

第1章 C语言概述

本章要点

本章主要介绍了C语言的发展简史、特点，C语言程序的构成及其书写格式和算法。

C语言作为高级语言具有语义简单、明确，容易理解和记忆等特点，又具有和其他高级语言不同的特点。C语言具有丰富的数据类型，和多样的运算符，使得它能够为我们提供各种复杂的数据结构，实现各种运算。

1. C语言是以函数为模块的结构化程序设计语言，符合现代编程风格的要求。
2. 一个C程序总是从 `main` 函数开始执行。
3. C语言书写格式自由，一行内可以写几个语句，一个语句也可以写成多行。
4. C语言程序是用小写字母书写。
5. 可以用 `/*.....*/` 对C程序中的任何部分作注释。
6. 算法在程序设计中是很重要的，通常有四种表示方式：自然语言、伪代码、流程图、计算机语言。
7. C语言程序必须经过编辑、编译、连接后方可运行。
8. 在 Turbo C 环境下常用的功能键：
Ctrl+F9 /*编译、连接和运行*/
Alt+F5 /*切换到屏幕*/
F6 /*取消亮条*/

实验1 C程序的运行环境和编辑一个C程序的方法

一、实验目的

1. 熟习 Turbo C 的上机过程，在 Turbo C 环境下学习如何编辑、编译、连接和运行一个C程序。
2. 通过运行简单的C程序，初步了解C源程序的特点。

二、实验要求

1. 预习 Turbo C 集成环境的基本操作过程。

2. 预习 C 语言编写程序的基本结构和方法。

三、实验内容

1. 进入 Turbo C 工作环境

- (1) 单击 Windows 桌面上的“开始”按钮，从菜单中选择“程序”，单击后在其下拉菜单中选择“MS-DOS”，屏幕上出现 MS-DOS 窗口。
- (2) 用 DOS 命令 `cd` 使当前目录改变为用户专用的子目录（例如 `C:\WANG`）。
- (3) 用 `dir` 命令检查用户目录中已有哪些文件。
- (4) 调用 Turbo C，一般情况下 Turbo C 所在的目录为：`C:\TC`。

调用 Turbo C 的命令为：

```
C:\TC>tc✓
```

此时屏幕上会出现 Turbo C 的工作窗口。

2. 熟悉 Turbo C 集成环境

- (1) 了解和熟悉编辑 (Edit) 窗口和信息 (Message) 窗口的作用。试一下能否向信息窗口输入程序。按功能键 F5，观察它有什么作用，再按一次 F5，再观察它有什么作用。按 F6，观察它有什么作用；再按一次 F6，再观察它有什么作用。
- (2) 按功能键 F10，并按回车键，观察“File”菜单，了解它们的作用和用法。用键盘上的“→”键，分别选中 Edit、Run、Compile、Project、Option、Break/watch，观察它们的菜单，大致上了解它们的作用（暂时不必要求全部深入了解，以后逐步熟习）。
- (3) 输入并运行一个简单的程序。

具体步骤：

- ① 先选择 File 菜单中的 New 命令并按回车键，使编辑窗口变成空白。
- ② 输入下面的程序

```
main()
{
    printf("This is a C program.\n");
}
```

- ③ 按功能键 F9 进行编译和连接，观察屏幕上显示的编译信息。如果出现“出错信息”，则找出原因并改正之，再进行编译。
- ④ 如果编译没有错误，按 Ctrl 和 F9 键运行程序，按 Alt 和 F5 键，切换到用户屏幕，观察分析运行结果。

3. 输入调试下面程序

- (1) 从键盘依次输入五个实数，计算并输出这五个实数的平均数。

```
#include "stdio.h"
main()
{ float a,b,c,d,e,av;
  printf("input a,,b,c,d,e:");
```

```
scanf("%f, %f, %f, %f, %f",&a,&b,&c,&d,&e);
av=(a+b+c+d+e)/5;
printf("av=%e\n",av);
}
```

具体要求:

- ① 在 Turbo C 编辑窗口输入该程序, 然后进行编译连接。如果在编译过程中有错误发生, 则仔细检查并修改程序, 再进行编译连接, 直到没有错误为止。
- ② 运行时按程序中要求的格式依次输入下列五个数:

12.4, -5.23, 20.78, 16.22, 3.14

(2) 有以下程序。

```
main()
{int a,b,c;
  printf("input a & b : ");
  scanf("%d,%d",&a,&b);
  c=min(a,b);
  printf("min=%d",c);
}

int min(int x, int y)
{  int z;
   if(x<y) z=x;
   else   z=y;
   return (z);
}
```

具体要求:

- ① 输入并调试源程序;
 - ② 掌握在 Turbo C 环境下的编辑、连接、运行过程。
- (3) 编写一个 C 程序, 输入 a, b, c 三个实数, 输出其中最大者。

具体要求:

- ① 在 Turbo C 环境下编辑该程序, 然后进行编译连接。
- ② 运行该程序, 检查你所编写的程序是否正确。
- ③ 画出流程图。

第 2 章 数据类型与运算规则

本章要点

C 语言的数据类型非常丰富,本章介绍了 C 语言的基本数据类型、C 语言的算术运算符、强制类型转换运算符、自加运算符(++)和自减运算符(--)、赋值运算符、逗号运算符,以及由这些运算符将运算数据连接起来而组成的表达式。这些是组成 C 语言程序语句的基本成分。

1. C 语言中的标识符是由字母、下划线打头,后跟字母、数字和下划线组成的字符序列。大小写是有区别的。
2. 不同数据类型占不同的内存长度,有不同的取值范围,需牢记。

基本类型的长度和取值范围:

类型	类型说明符	字节	取值范围
字符型	char	1	字符集
基本整型	int	2	-32768~32767
短整型	short int	2	-32768~32767
长整型	long int	2	-2147483648~2147483647
无符号型	unsigned	2	0~65535
无符号长整型	unsigned long	2	0~4294967295
单精度实型	float	4	3.4e-38~3.4e+38
双精度实型	double	8	1.7e-308~1.7e+308

3. 整型常量有十进制、八进制(以 0 打头)、十六进制(以 0x 打头)和长整型(l 或 L 后缀)四种表示形式;实型常量有小数和指数两种表示形式;字符常量有单个字符和转义字符两种;还有字符串常量和符号常量。
4. 求余(%)运算只适用于整型; ++i(--i)是先增(减)值后使用, i++(i--)是先使用再增(减)值。
5. 单目运算符、三目运算符和赋值运算符是从右向左结合;其余的运算按常规从左向右结合;括号可以改变运算的优先次序。
6. 数据类型转换可以采用自动类型和强制类型两种形式。
 - (1) 自动类型 在不同类型混合运算中,由系统自动转换,由少字节向多字节类型转换。不同的量相互赋值时也由系统自动转换,把赋值号右边的类型转换为左边的类型。
 - (2) 强制类型 由强制转换类型说明符完成转换。

7. 语句的分类

C 语言的语句可分为两大类：说明语句和可执行语句。

(1) 说明语句：对程序中所使用的变量及其属性进行说明（又称定义）。

(2) 可执行语句：完成程序功能的语句。根据表现形式和功能，又分为五类：表达式语句、函数调用语句、控制转移语句、复合语句、空语句，这五类语句是编写 C 语言的基本语句。

8. 表达式

关系表达式和逻辑表达式是两种重要的表达式，主要用于条件执行和循环执行的判断，分别由 6 个关系运算符（<、<=、>、>=、==、!=）和 3 个逻辑运算符（&&、||、!）组成。

实验 2 数据类型、运算符和表达式

一、实验目的

1. 掌握 C 语言数据类型，熟悉如何定义一个整型、字符型、实型变量，以及对它们赋值的方法，同时要了解以上类型数据输出时所用的格式转换符。
2. 掌握 C 语言的算术运算符、自加运算符（++）和自减运算符（--）以及赋值运算符（=）的使用，并且学会如何使用这些运算符正确书写表达式。

二、实验要求

1. 预习教材第 2 章内容。
2. 进一步练习 C 程序的编辑、编译、连接与运行的过程。

三、实验内容

1. 输入并运行下面的程序。

```
main()
{   int  m,n;
    m=120;
    n=112;
    printf("%d\t%d\n",m,n);
}
```

具体要求：

- ① 记下程序运行的结果。
- ② 在程序的最后增加一句：

```
printf("%c\t%c\n",m,n);
```

再运行程序，并分析结果。

③ 将第 2 行改为:

```
char m,n;
```

再运行程序, 并观察结果。

④ 将第 3 行和第 4 行改为:

```
m='x';
```

```
n='p';
```

再运行程序, 并观察结果。

⑤ 将第 3 行和第 4 行改为:

```
m=x;
```

```
n=p;
```

再运行程序, 并观察结果。如果产生了错误, 说明是什么引起的?

⑥ 再将第 3 行和第 4 行改为:

```
m=280; /*用大于 255 的整数*/
```

```
n=320;
```

再运行程序, 并观察分析运行结果。

2. 输入并运行下面的程序。

```
main()
{
    char c1='x', c2='y', c3='z', c4='\101', c5='\102';
    printf("x%cy%cz%t\n", c1, c2, c3);
    printf("\tb%%c", c4, c5);
}
```

具体要求:

上机前先分析程序, 并在实验报告上写出程序执行的结果, 上机后再将二者对照检查。

3. 输入并运行下面的程序。

```
main()
{
    int a,b;
    unsigned c,d;
    long e,f;
    a=100; b=-100;
    e=50000;
    f=32767;
    c=a;
    d=b;
    printf("%d,%d\n", a,b);
    printf("%u,%u\n", a,b);
    printf("%u,%u\n", c,d);
    c=a=e;
```

```
d=b=f;
printf(“%d,%d\n”,a,b);
printf(“%u,%u\n”,c,d);
}
```

具体要求（请对照程序运行结果分析）：

- ①将一个负整数赋给一个无符号的变量，会得到什么结果？画出它们在内存中的表示形式。
 - ②将一个大于 32767 的长整型数据赋给整型变量，会得到什么结果？画出它们在内存中的表示形式。
 - ③将一个长整型数据赋给无符号变量，会得到什么结果？（分别考虑该长整数的值大于或等于 65535 以及小于 65535 的情况）。画出它们在内存中的表示形式。
- 读者可以改变程序中各变量的值，以便比较。例如：a=65580,b=-40000,e=65535,f=65580。

4. 输入下面的程序。

```
main()
{int a,b,c,d;
  a=12;
  b=-5;
  c=++a;
  d=b++;
  printf(“%d,%d,%d,%d\n”,a,b,c,d);
}
```

具体要求：

- ① 在运行程序之前阅读程序并分析程序运行的结果（将其写实验报告上）。
- ② 运行程序，并与你所分析的结果对照一下。
- ③ 将第 5、6 行改为：

```
c=a++;
d=++b;
```

再运行，并注意观察结果有何不同？

- ④将程序改为：

```
main()
{ int a,b;
  a=10;
  b=-8;
  printf(“%d,%d\n”,a++,b++);
}
```

运行程序。

- ⑤在④的基础上，将 printf 语句改为：

```
printf(“%d,%d\n”,++a,++b);
```

运行程序，并将结果与上题对照。

⑥再将 printf 语句改为：

```
printf("%d,%d,%d,%d\n",a,b,a++,b++);
```

运行程序。

⑦将程序改为：

```
main()
{ int a,b, c=0,d=0;
  a=10;
  b=8;
  c+=a++;
  d=--b;
  printf("a=%d,b=%d,c=%d,d=%d\n",a,b,c,d);
}
```

运行程序。

第 3 章 程序控制结构

本章要点

一、数据的输入/输出

- 1. C 语言中，所有数据的输入/输出都是由库函数完成的。
 - 2. 字符输入/输出函数 `getchar`、`putchar`，只能接受、显示单个字符，使用时必须在源程序清单的开头加上包含命令“`#include <stdio.h>`”。
 - 3. 格式输入/输出函数 `scanf()`、`printf()`，使用时输入/输出函数的格式由格式控制字符串指定，各输入、输出项在数量、类型上必须和格式字符串保持一致，在源程序清单的开头可以加上(也可以省略)包含命令：“`#include <stdio.h>`”。
- 输入/输出格式字符以及所控制的数据类型、形式、方法见下表：

格式输入字符表

格式字符	数据对象	输入形式	数据输入方法
%md	int、short、 unsigned int、 unsigned short、	十进制整数	无 m 按实际位数输入 有 m 输入 m 位， 不足 m 则跟回车键
%mo		八进制整数	
%mx		十六进制整数	
%mld	long unsigned long	十进制长整数	无 m 按实际位数输入 有 m 输入 m 位， 不足 m 则跟回车键
%mlo		八进制长整数	
%mlx		十六进制长整数	
%mlf	float double	十进制实数	无 m 按实际位数输入， 有 m 输入 m 位，不足 m 则跟回车键
%mle		十进制指数	
%mc	char	单个字符	无 m 仅取单个字符 有 m 输入 m 位，仅取第一个字符
%ms	字符串	一串字符	无 m 取回车或空格前若干个字符 有 m 仅取前 m 字符

格式输出字符表

格式字符	数据对象	输出形式	数据输出方法
%-md	int、float unsigned int unsigned short char	十进制整数	无 m 按实际位数输出。有 m 输出 m 位； 超过 m 位，按实际位数输出；不足 m 位 补空格。 无-右对齐（左补空格）。 有-左对齐（右补空格）
%-mo		八进制整数	
%-mx		十六进制整数	
%-mu		无符号整数	
%-mld	long unsigned long	十进制长整数	无 m 按实际位数输出。有 m 输出 m 位； 超过 m 位，按实际位数输出；不足 m 位 补空格。无-右对齐（左补空格）。 有-左对齐（右补空格）
%-mlo		八进制长整数	
%-mlx		十六进制长整数	
%-mlu		无符号长整数	
%-m.nf	float double	十进制实数	无 m 按实际位数输出。有 m 输出 m 位； 超过 m 位，按实际位数输出；不足 m 位 补空格。 无-右对齐（左补空格）。 有-左对齐（右补空格）
%-m.ne		八进制指数	
%-g		自动选取 f 或 e 中 宽度的格式	
%-mc	char int short	单个字符	无 m 输出单个字符。 有 m 输出 m 位，补空格 无-右对齐（左补空格） 有-左对齐（右补空格）
%-m.ns	字符串	一串字符	无 m.n 按实际字符串输出全部字符 有 m.n 仅输出前 n 个字符，补空格 无-右对齐（左补空格） 有-左对齐（右补空格）

使用 scanf () 函数给变量赋值时，应书写变量的地址，如：scanf(“%d",&a)， “&” 为取地址运算符。C 语言规定空格、Tab、回车或非法数据都是数据之间的间隔符号，不作为有效字符。

二、三种控制结构

结构化程序的标准是任何程序均由顺序、分支和循环三种基本模块组成。不允许随意跳转。任何程序模块不论大小，应只有一个入口，一个出口，没有死语句，没有死循环。

1. 顺序结构

该结构程序设计步骤：输入数据→计算→输出。

2. 选择结构(分支结构)

该结构程序设计分三种情况：

- (1) 单分支结构，用来解决“依据一个条件决定某个操作做还是不做”的问题（用 if 语句实现）。
- (2) 双分支结构，用来解决“依据一个条件，从两个操作中选择一个操作来做”的问题（用 if-else 语句实现）。
- (3) 多分支结构，用来解决“依据 n 个条件，从 n+1 个操作中选择一个操作来做”的问题（用 if-else-if 和 switch 语言实现）。

3. 循环结构

- (1) **while** 语句（当型循环语句）：当某个条件成立(条件值非 0) 时，反复执行某个操作；
- (2) **do-while** 语句（直到型循环语句）：反复执行某个操作，直到某个条件不成立（条件值为 0）时为止；
- (3) **for** 语句（次数型循环语句）：由循环的初值，终值和步长（增量）控制的有限次循环。
- (4) 循环体中可以是单语句，也可以是多个语句，甚至可以是空语句。若是多个语句，则必须用花括号 { } 括起，构成复合语句。
- (5) 在循环结构中，利用 **break** 语句可以从循环体内跳出循环体，即提前结束循环；利用 **continue** 语句可以结束本次循环，接着执行下一次是否执行循环的判定。
- (6) 使用嵌套循环结构时，外层的循环结构一定要完全包含内层循环，决不允许交叉嵌套。

实验3 顺序程序设计

一、实验目的

1. 掌握 C 语言中基本语句的使用及计算，掌握顺序程序设计的一般方法。
2. 掌握字符输入函数 **getchar()** 和字符输出函数 **putchar** 的用法。
3. 掌握格式输入函数 **scanf()** 和格式输出函数 **printf()** 的用法。
4. 正确使用各种输入/输出格式。

二、实验要求

1. 预习 C 语言的基本结构和基本数据类型（**char**、**int**、**short**、**long** ...）。
2. 预习输入/输出函数（**getchar**、**putchar**、**scanf()**、**printf()**）的调用格式与功能。
3. 预习 Turbo C 的基本操作方法，学会保存、修改源程序。

三、实验内容

1. 编写 C 程序，按下列公式计算并输出 S 的值。其中 a 和 b 的值由键盘输入。

$$S = \frac{2ab}{(a+b)^2}$$

源程序清单：

```
main
{
    int a,b;
```