

 普通高等院校“十二五”规划教材

# C语言程序设计 实验指导与习题解答

梅 毅 赵金萍 吴赞婷 主 编  
兰长明 熊 婷 副主编



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

普通高等院校“十二五”规划教材

# C 语言程序设计实验指导与 习题解答

梅 毅 赵金萍 吴贇婷 主 编

兰长明 熊 婷 副主编

**中国铁道出版社**  
CHINA RAILWAY PUBLISHING HOUSE

---

## 内 容 简 介

本书是《C 语言程序设计》的配套上机指导书,目的是使学生在“C 语言程序设计”这门课程的过程中,掌握更多重要的上机编程知识,为自己独立完成应用 C 语言编写结构化程序打下坚实的基础。

本书共提供 24 个实验,其中,C 语言基础知识 4 个;顺序、选择和循环结构程序设计知识 5 个;编译预处理知识 1 个;数组知识 3 个;函数知识 3 个;指针知识 2 个;结构体、位运算和文件应用知识 3 个;C 语言程序综合应用知识 3 个。此外,本书还提供了主教材各章课后习题参考答案。

本书适合作为“C 语言程序设计”课程的实验指导书,也可作为“C 语言程序设计”课程上机练习的自学用书,以及 C 语言初学者的练习用书。

### 图书在版编目(CIP)数据

C 语言程序设计实验指导与习题解答 / 梅毅, 赵金萍,  
吴赞婷主编. — 北京: 中国铁道出版社, 2014.8  
普通高等院校“十二五”规划教材  
ISBN 978-7-113-18743-9

I. ①C… II. ①梅… ②赵… ③吴… III. ①C 语言—  
程序设计—高等学校—教学参考资料 IV. ①TP312  
中国版本图书馆 CIP 数据核字(2014)第 167118 号

书 名: C 语言程序设计实验指导与习题解答

作 者: 梅 毅 赵金萍 吴赞婷 主编

策 划: 曹莉群

读者热线: 400-668-0820

责任编辑: 周海燕 彭立辉

封面设计: 刘 颖

封面制作: 白 雪

责任校对: 汤淑梅

责任印制: 李 佳

出版发行: 中国铁道出版社(100054, 北京市西城区右安门西街 8 号)

网 址: [http:// www.51eds.com](http://www.51eds.com)

印 刷: 三河市宏盛印务有限公司

版 次: 2014 年 8 月第 1 版 2014 年 8 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 10 字数: 251 千

印 数: 1~3 000 册

书 号: ISBN 978-7-113-18743-9

定 价: 22.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

C 语言是国内外广泛使用的一种计算机语言。很多高校都开设了“C 语言程序设计”课程。C 语言是一门实践性很强的计算机语言,在学习过程中,检查学习效果的标准,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,学习者一定要重视实践环节,多上机练习。为了更好地帮助学生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,我们编写了《C 语言程序设计实验指导与习题解答》。本书是《C 语言程序设计》(熊婷、兰长明主编,中国铁道出版社出版)的配套用书,采用的上机环境为 Visual Studio C++ 6.0。

本书包括两大部分:一是与主教材各章节对应的实验,每个实验由实验目的与要求、知识要点、实验示例和实验内容组成;二是主教材各章课后习题答案,为学生课后练习提供参考。

本书提供 24 个实验(建议每个 2 学时),每个实验中,都包括对应实验的实验目的与要求、知识要点、实验示例和实验内容。其中,C 语言基础知识 4 个;顺序、条件和循环结构程序设计知识 5 个;编译预处理知识 1 个;数组知识 3 个;函数知识 3 个;指针知识 2 个;结构体、位运算和文件应用知识 3 个;C 语言程序综合应用知识 3 个。

本书由南昌大学科技学院梅毅、赵金萍、吴赞婷任主编,兰长明、熊婷任副主编,其中,实验 1、8、9、19 由吴赞婷编写,实验 2、3、17、18 由赵金萍编写,实验 4、5、10、11、12、13 由熊婷编写,实验 6、7、14、15、16 由兰长明编写,实验 20、21、22、23、24 由梅毅编写,全书由熊婷、梅毅统稿。另外,在编写过程中,张炘、王钟庄、邹璇、刘敏、范晰、罗少彬、邓伦丹、罗丹、汪伟、周权来、李昆仑、汪滢、张剑、罗婷等老师做了大量的辅助工作,并提出了许多宝贵意见;中国铁道出版社编辑、南昌大学科技学院的各级领导、计算机系全体教职员给予了大力支持,在此一并表示衷心感谢!

本书由从事多年“C 语言程序设计”课程教学的一线老师组织编写,但由于时间仓促,编者水平有限,仍难免存在疏漏与不妥之处,敬请读者批评指正。

编 者

2014 年 5 月

## 第一部分 实 验

|                                      |     |
|--------------------------------------|-----|
| 实验 1 熟悉 C 语言的运行环境.....               | 1   |
| 实验 2 数据类型应用.....                     | 5   |
| 实验 3 运算符与表达式.....                    | 9   |
| 实验 4 格式输入/输出应用.....                  | 13  |
| 实验 5 顺序结构设计.....                     | 19  |
| 实验 6 if 选择结构程序设计与应用.....             | 25  |
| 实验 7 switch 选择结构以及 if 嵌套程序设计与应用..... | 30  |
| 实验 8 C 语言单循环程序设计.....                | 35  |
| 实验 9 C 语言多循环程序设计.....                | 39  |
| 实验 10 编译预处理应用.....                   | 43  |
| 实验 11 一维数组的应用.....                   | 48  |
| 实验 12 二维数组的应用.....                   | 54  |
| 实验 13 字符数组的应用.....                   | 59  |
| 实验 14 函数基础.....                      | 64  |
| 实验 15 函数应用 1.....                    | 69  |
| 实验 16 函数应用 2.....                    | 74  |
| 实验 17 指针与函数以及指针与数组.....              | 79  |
| 实验 18 指针与字符串.....                    | 88  |
| 实验 19 结构体与共用体的应用.....                | 93  |
| 实验 20 位运算应用.....                     | 98  |
| 实验 21 文件应用.....                      | 102 |
| 实验 22 综合实验一.....                     | 108 |
| 实验 23 综合实验二.....                     | 114 |
| 实验 24 综合实验三.....                     | 124 |

## 第二部分 主教材课后习题答案

|                   |     |
|-------------------|-----|
| 第 1 章 课后习题答案..... | 131 |
| 第 2 章 课后习题答案..... | 132 |
| 第 3 章 课后习题答案..... | 133 |

|                    |     |
|--------------------|-----|
| 第 4 章 课后习题答案.....  | 135 |
| 第 5 章 课后习题答案.....  | 137 |
| 第 6 章 课后习题答案.....  | 139 |
| 第 7 章 课后习题答案.....  | 141 |
| 第 8 章 课后习题答案.....  | 145 |
| 第 9 章 课后习题答案.....  | 147 |
| 第 10 章 课后习题答案..... | 150 |
| 第 11 章 课后习题答案..... | 152 |
| 第 12 章 课后习题答案..... | 153 |

# 第一部分 实 验

## 实验 1 熟悉 C 语言的运行环境

### 【实验目的与要求】

- (1) 了解 Visual C++ 6.0 (以下可简称 VC 6.0) 的菜单结构, 熟练地输入和修改源程序。
- (2) 掌握 C 语言程序的编译、连接和运行过程。
- (3) 通过运行简单的 C 语言源程序, 掌握 C 语言的上机步骤。
- (4) 掌握在 Visual C++ 6.0 环境下检查错误的方法。

### 【知识要点】

复习主教材第 1 章, 具体掌握的知识点如下:

- (1) C 语言是一种结构化程序设计语言, 既具有高级语言的特点, 也具有低级语言的特点, 可以用来编写系统软件, 还可以用来编写应用软件。
- (2) C 程序中除了字符型数据外, 其他所有成分都只能由字符集中的字符构成; 而变量名、函数名、数组名等都是用标识符来命名。
- (3) 在使用库函数或者其他文件中的函数时, 要注意使用相应的文件包含命令。
- (4) 在 Visual Studio C++ 6.0 环境下编辑与运行 C 程序的步骤:
  - ① 启动 Visual Studio C++ 6.0。
  - ② 创建一个新的 C 程序文件。
  - ③ 编辑 C 程序代码。
  - ④ 保存 C 程序文件。
  - ⑤ 源程序文件的编译、连接。
  - ⑥ 若程序出现错误, 按照“消息”窗口给出的错误提示进行修改, 然后重复步骤⑤, 直至错误个数为 0, 警告个数为 0。
  - ⑦ 运行可执行程序, 如果发现运行错误, 重复步骤⑥, 直至结果出来。

### 【实验示例】

1. 求 a 与 b 两数值的和。

(1) 分析: 很明显本题的数值是 a 和 b, 但是不要忽略另一个数值就是两数的和, 所以要定义三个变量。先输入 a 和 b 的值, 然后再求和。

(2) 编程代码:

```
#include<stdio.h>
main()
{
    int a,b,sum;
    scanf("%d %d",&a,&b);
    sum=a+b;
    printf("sum=%d\n",sum);
}
```

这时若程序出现错误（或警告），屏幕下方将显示出错误的个数以及错误的类型，如图 1-1 所示。

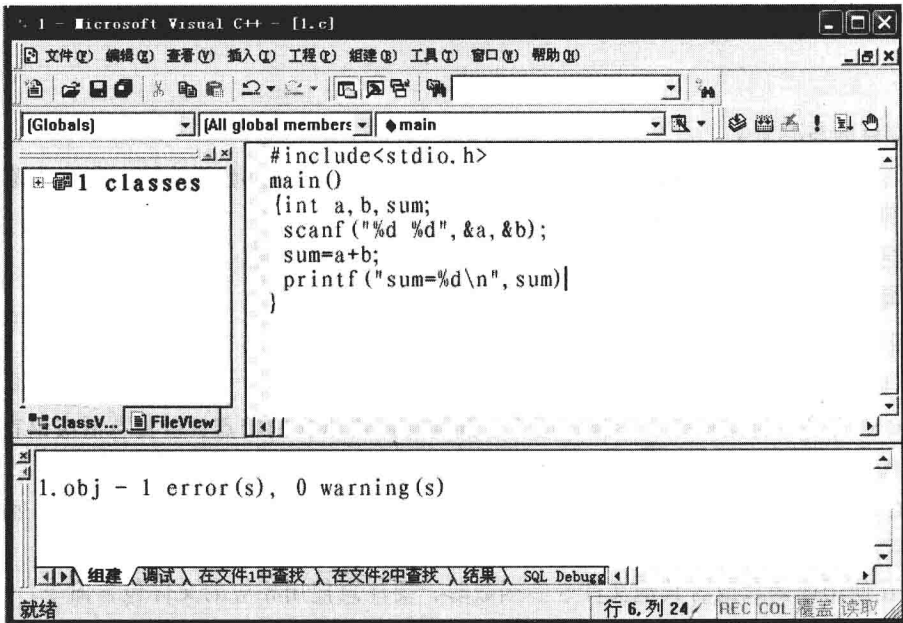


图 1-1 程序出现错误

检查出错误的位置，选中并修改，修改完成后，选择菜单栏中的“组建”→“组建”命令，没有错误后，选择“组建”→“执行”命令，弹出结果窗口。

(3) 执行结果如图 1-2 所示。

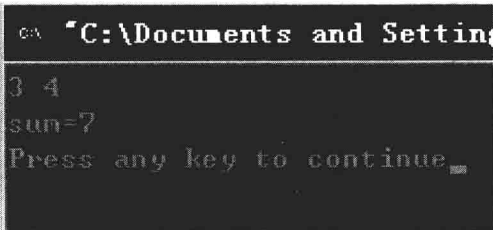


图 1-2 实验示例 1 的运行结果

(4) 注意事项：在程序的第 5 行“scanf("%d %d",&a,&b);”中，两个%d之间用空格分开，在输入的时候也要注意在相应的位置输入空格。

## 2. 求某整数的绝对值。

(1) 分析：求绝对值分为两种情况，第一，若该数是正整数或 0，那么绝对值便是该数本身；第二，若该数是负整数，那么绝对值要再对该数做一次负号运算。

(2) 代码如下：

```
#include<stdio.h>
void main()
{
    int a;
    scanf("%d",&a);
    if(a>=0)
        printf("%d的绝对值是%d\n",a,a);
    else if(a<0)
        printf("%d的绝对值是%d\n",a,-a);
}
```

(3) 程序运行结果如图 1-3 所示。



图 1-3 实验示例 2 程序运行结果

(4) 注意事项：求某一个数的绝对值，结果必须是正数。

### 【实验内容】

根据实验示例，完成以下实验内容：

1. 了解 VC 6.0 环境的组成，进入和退出 VC 6.0 环境。
2. 编写如下程序，并输出运行结果。

```
#include<stdio.h>
main()
{
    printf("This is a C Program\n");
    printf("I am a student\n");
}
```

3. 编写一个 C 程序，输入 a、b、c 三个值，输出最大值。
4. 改正下列程序中的错误，并上机验证。

```
#include<stdio.h>
void main()
{
    float a,b,area; /*变量定义*/
    a=1.9
    b=3.6
```

```
    area=abb;  
    printf("a=%f,b=%f,面积=%f\n",a,b,area);  
}
```

5. 观察下面程序的功能,运行出来的结果是什么。程序语法上并没有错误,要怎样修改才能得到正确的结果?请上机验证。

```
#include<stdio.h>  
void main()  
{  
    int a=1,b=2,c;  
    c=a/b;  
    printf("%d\n",c);  
}
```

## 实验 2 数据类型应用

### 【实验目的与要求】

- (1) 掌握常量的使用。
- (2) 掌握整型、实型、字符型变量的定义及使用方法。
- (3) 进一步熟悉 C 程序的编辑、编译、连接和运行操作。

### 【知识要点】

复习主教材第 2 章的 2.1 C 语言中的数据类型、2.2 常量与变量、2.3 整型数据、2.4 实型数据、2.5 字符型数据等 5 节内容。应掌握的知识点如下：

- (1) C 语言中常量是程序在运行过程中不可以改变的量，包括字面常量与符号常量。
- (2) C 语言中变量是程序在执行过程中，值可以改变的量，其实就是可以不断被赋值的量。变量对应一个固定的存储空间，变量被改变的实质就是用新值覆盖旧值。
- (3) 变量必须先定义后使用。

### 【实验示例】

1. 输入并运行下面的程序。

(1) 分析：通过程序展示常量与变量的使用。

(2) 编程代码：

```
#define N 3.5
#include<stdio.h>
main()
{
    float a;
    a=2.5;
    printf("a=%f\n",a);
    a=N;
    printf("a=%f\n",a);
    a=5.5;
    printf("a=%f\n",a);
}
```

(3) 打开 Visual C++ 6.0，输入代码，然后编译→连接→运行，结果如图 2-1 所示。

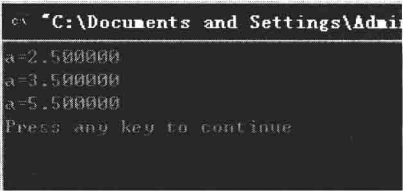


图 2-1 实验示例 1 程序运行结果

(4) 注意事项：本例中定义了一个符号常量 N，一个实型变量 a，在程序中，不断对变量 a 进行赋值，这样做是允许的，因为变量本身就是可以不断被改变的，只是保存的是最后一次被赋的值。可以从本例中看到，第一次变量 a 赋值 2.5，所以紧跟其后的输出结果是 a=2.500000，被再次赋值 N 时，输出结果就是 3.500000，而最后保存的是最后一次的值 5.500000。这里希望大家注意，只有变量才可以被赋值。

如果程序变成如下程序：

```
#define N 3.5
#include<stdio.h>
main()
{
    float a;
    a=2.5;
    2=45;                /*这里将 45 赋给字面常量 2,错误*/
    N=5.5;               /*这里使用字面常量 5.5 赋值给字符常量 N, 错误*/
    printf("a=%f\n",a);
}
```

运行程序，将出现如图 2-2 所示的错误提示。

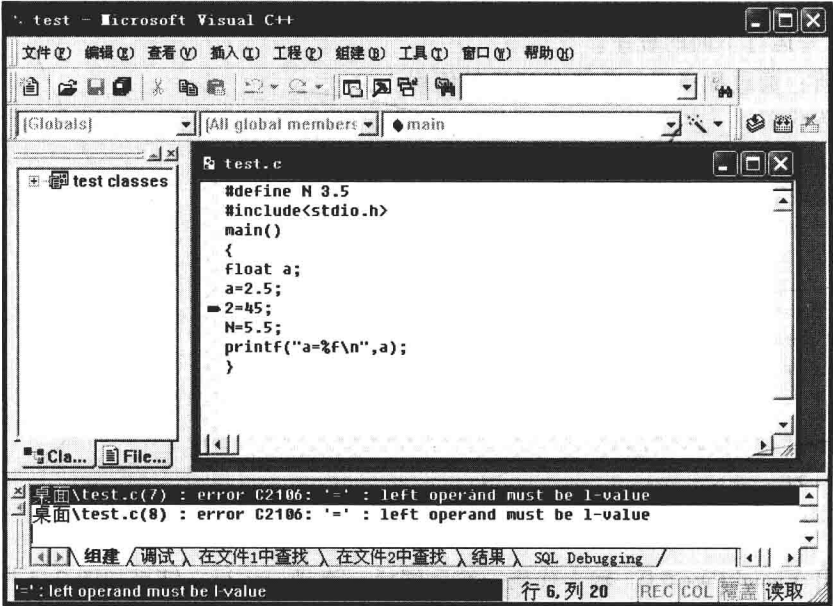


图 2-2 常量被赋值提示的错误结果

因此，不论是符号常量还是字面常量，在程序中都不能被改变，只有变量才可以被赋值，

可以把对应类型的常量（包括符号常量、字面常量）赋值给变量。

2. 分析下列程序，并上机运行。

编程代码：

```
#include<stdio.h>
main()
{
    int a1,a2,a3;
    float b1,b2;
    char c1,c2;
    a1=123;
    a2=0123;
    a3=0x123;
    b1=1.23;
    b2=1.23e2;
    c1='A';
    c2='\101';
    printf("a1=%d, a2=%d, a3=%d\n",a1,a2,a3);
    printf("b1=%f, b2=%f\n",b1,b2);
    printf("c1=%c,c2=%c\n",c1,c2);
}
```

（1）分析：本例主要涉及数据类型的定义，以及不同数据类型的常量有不同的表示形式。对这些不同表示形式的使用需要注意表示的合法性以及表示的确定值。

（2）打开 VC 6.0，输入代码，然后编译→连接→运行，结果如图 2-3 所示。

```
C:\Documents and Settings\Admini
a1=123, a2=83, a3=291
b1=1.230000, b2=123.000000
c1=A,c2=A
Press any key to continue.
```

图 2-3 实验示例 2 的程序运行结果

（3）注意事项：本例中定义了 3 个整型变量 a1、a2、a3 和 2 个实型变量 b1、b2 以及 2 个字符型变量 c1、c2，并对这 7 个变量进行了赋值。在对变量进行赋值时，一定要注意以下几点：

① 变量的数据类型与所赋值的常量属于同一数据类型。例如，a1、a2、a3 是整型变量，只能正确保存整型数据，但是必须注意这些变量的存储范围。所以，如果有 a1=2.5；编译系统其实并不会判断为错误，只是 a1 只能得到 2，不能完全保存 2.5 这个数值。

② 因为整型、实型及字符型常量都存在不同的表示形式，所以在给变量赋值时一定要注意所赋值的表示是否合法，值是多少。例如，程序中 a1=123;是十进制的 123，而 a2=0123 则为八进制的数，a3=0x123 则为十六进制的数。请注意这些不同的表示形式具体含义。另外，在输出这些数值时，结果和当初赋值的数似乎不同，这一点涉及格式化输入/输出的内容，在之后章节将详细讲解，这里只需要知道，“%d”对应的是十进制有符号整数的输入/输出，“%f”对应

的是一般形式的单精度实型数的输入/输出, 输出时默认 6 位小数, “%c” 对应的是一般形式的字符数据的输入/输出。

### 【实验内容】

根据实验示例, 完成以下实验内容:

1. 上机验证主教材例 2-3~例 2-7。
2. 使用 VC 6.0 调试下列两段程序, 分析语句含义, 修改其中错误并最终正确执行, 查看输出结果。

代码段 (1):

```
#include<stdio.h>
#define PI 3.14;
void mian()
{
    int a
    float b;
    a=3;
    4=a;
    b=pi;
    PI=b;
    printf("a=%d\n",a);
    printf("b=%f\n",b);
}
```

代码段 (2):

```
#include<stdio.h>
void main()
{
    int a,b,c;
    float x;
    char s;
    a=78;
    b=078;
    c=0xeg;
    x=1.23e2.5;
    s="a";
    printf("a=%d,b=%d,c=%c\n",a,b,c);
    printf("b=%d\n",b);
    printf("c=%d\n",c);
}
```

3. 编程从键盘输入两个整数, 输出它们的和、差、积、商。
4. 编程从键盘输入两个字符, 输出它们的大小写两种形式, 并输出与之对应的 ASCII 码值。

## 实验 3 运算符与表达式

### 【实验目的与要求】

- (1) 掌握不同数据类型之间的强制转换。
- (2) 掌握算术运算符、关系运算符、逻辑运算符以及其他运算符的运算规则。
- (3) 掌握表达式的组成及使用。

### 【知识要点】

复习主教材第 2 章的 2.6 运算符和表达式、2.7 运算符的优先级与类型的强制转换。具体掌握的知识点如下：

(1) 算术运算符的运算规则，特别注意除 (/) 和取余 (%) 运算。除法分为整除和精确除，而取余运算符只能作用在整型数据上。

(2) 使用关系运算符和逻辑运算符来表示数学中的关系表达式。特别注意等于比较运算符“==”与赋值运算符“=”的区别。

(3) 自增运算符(++)、自减运算符(--)、条件运算符(?:)以及逗号运算符(,)的使用，特别注意自增自减的使用。

(4) 强制类型转换的使用，包括隐式转换和显示转换。

### 【实验示例】

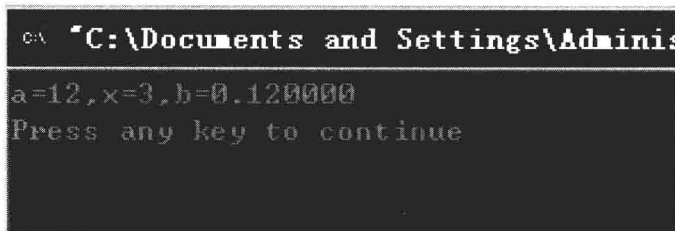
1. 上机验证下列程序，分析代码含义并查看运行结果。

编程代码：

```
#include<stdio.h>
main()
{
    int a=123;
    int x;
    float b=1.2;
    x=a%10;
    a=a/10;
    b=b/10;
    printf("a=%d,x=%d,b=%f\n",a,x,b);
}
```

(1) 分析：本例主要涉及了算术运算符的使用，在使用“/”时，如果两个操作数同为整型数，则为整除（结果只保留结果的整数部分，小数部分舍弃），如果两操作数不同时为整型数时，所得的结果为准确结果，即包括整数及小数部分（如果有小数）。

(2) 打开 VC 6.0, 输入代码, 然后编译→连接→运行, 结果如图 3-1 所示。



```

C:\Documents and Settings\Adminis
a=12,x=3,b=0.120000
Press any key to continue

```

图 3-1 实验示例 1 程序运行结果

(3) 注意事项: 对照程序的结果可以发现,  $a=a/10$ ; 执行的是整除, 因为除号的上下两个操作数  $a$  和  $10$  都是整数, 所以为整除, 结果只保留所得结果的整数部分, 而  $b=b/10$ ; 执行的是准确的除法, 因为除号的上下两个操作数为  $b$  和  $10$ ,  $b$  是实型的单精度数, 所以不满足上下两个数同时为整数的条件, 则执行的就是准确的除法。另外, 在程序中使用到  $x=a\%10$ ; 这里取余运算符的操作数都为整型数据, 是合法的, 如果程序里出现  $x=b\%10$ ; 则会出现错误, 因为取余操作只能针对整型数据。

2. 分析下列程序代码, 并上机执行, 查看结果。

编程代码:

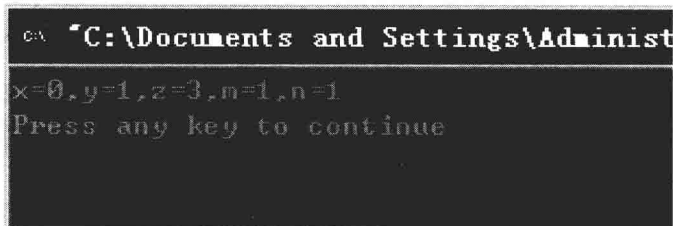
```

#include<stdio.h>
main()
{
    int a=3,b=4,c=5;
    int x,y,z,m,n;
    x=(c>b>a);
    y=(c>b&&b>a);
    z=(a=3);
    m=(a==3);
    n=a+b<c*a-b;
    printf("x=%d,y=%d,z=%d,m=%d,n=%d\n",x,y,z,m,n);
}

```

(1) 分析: 下面代码中使用几个整型变量保存右边表达式的值, 有关系表达式和赋值表达式。在进行关系表达式运算时, 注意与数学里的表达式的区别。另外, 因为关系运算符的优先级都低于算术运算符的优先级, 所以  $n=a+b<c*a-b$  等价于  $n=((a+b)<(c*a-b))$ 。

(2) 打开 VC 6.0, 输入代码, 然后编译→连接→运行, 结果如图 3-2 所示。



```

C:\Documents and Settings\Administ
x=0,y=1,z=3,m=1,n=1
Press any key to continue

```

图 3-2 实验示例 2 程序运行结果

## (3) 注意事项:

① 在 C 语言中,数据类型没有逻辑型,逻辑的真和假用数字 1 和 0 来表示,所以在本例程序中关系表达式的值都只有 0 或 1。用非 0 来表示真,用 0 表示假。

② 数学里的关系表达式  $c > b > a$  在 C 语言中应该表示成  $c > b \&\& b > a$ ,对照程序的运行结果可以得知,当  $a=3, b=4, c=5$  时,  $x=(c > b > a)$ ;  $x$  得到的结果却为假 0,这和数学里面的意义不同。

③ 程序中的  $z$  和  $m$  的区别是,  $z$  是获得赋值表达式  $a=3$  的值,也就是 3,而  $m$  则是得到关系表达式  $m==3$  的值,是用  $m$  与 3 进行比较,如果相等,表达式的值为真,即 1。所以,要注意 “=” 与 “==” 的区别。

3. 分析下面程序代码含义,上机运行并得到结果。

编程代码:

```
#include<stdio.h>
main()
{
    int a,b,m,x=1.2;
    float y;
    y=(x+13.8)/5;
    m=(int)y%2;
    a=3<2?3:4>3?4:3;
    b=a++,++m,++x;
    printf("x=%d,y=%f,m=%d,a=%d,b=%d\n",x,y,m,a,b);
}
```

(1) 分析:代码中出现了多次的强制类型转换,有些是隐式转换,有些则为显式的转换。另外,条件运算符的结合方向是右结合而逗号运算符的结合方向为左结合,并且逗号的优先级最低。

(2) 打开 VC 6.0,输入代码,然后编译→连接→运行,结果如图 3-3 所示。

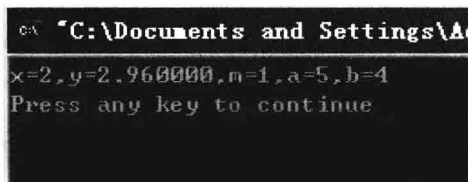


图 3-3 实验示例 3 程序运行结果

## (3) 注意事项:

① 在本例中,出现了多次隐式类型转换,  $\text{int } x=1.2$ ; 是给一个整型的变量  $x$  赋一个实型的常量 1.2,这是隐式转换的第一种形式,在赋值号的左右两边出现数据类型不一致的时候,无条件将右边转换成左边的变量类型,并且不是四舍五入的转换。第二个出现隐式转换的地方是  $y=(x+13.8)/5$ ; 这里出现了在一个运算符的两侧出现不同数据类型数据的时候,会发生类型的隐式转换,首先将  $x$  的值转换成  $1.0+13.8$ ,然后用结果与 5.0 进行除,即  $14.8/5.0$ ,所以得到的结果为 2.96。

② 本例程序中  $m=(\text{int})y\%2$ ; 使用了显示类型转换,这里将  $y$  的值转换成  $\text{int}$  型。这里需要注意写法,不要把  $(\text{int})y$  写成  $\text{int}(y)$ 。

③ 程序中使用了条件运算符,并且出现了多个条件运算符的情况,这时候应该注意条件运算符的结合方向是从右往左的。