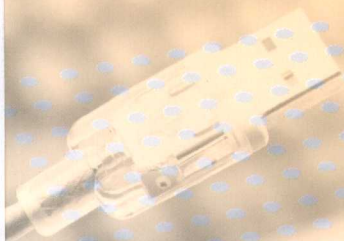


普通高校本科计算机类“十二五”规划教材

# C语言程序设计实践教程

编 著 韩立毛 徐秀芳 董 琴




 南京大学出版社

普通高校本科计算机类“十二五”规划教材

# C语言程序设计实践教程

编 著 韩立毛 徐秀芳 董 琴

TP312C-43  
944

 南京大学出版社

## 内容简介

本书结合 C 语言程序设计的特点以及初学者学习时的难点,构建了实验操作、模拟实战以及工程实训为一体的实践教学体系,实现了从以计算机语言为主线的体系结构向以问题为主线的体系结构上的转变,把程序设计的学习从语法知识学习提高到解决问题的能力培养上。全书共分为 3 个部分,第 1 部分为实验操作,共包括 16 个实验,每个实验包含实验目的、相关知识点、实验内容、具体要求以及程序设计提示。第 2 部分为模拟实战,共包含 20 套全国计算机等级考试上机考试模拟题。第 3 部分为工程实训,共包含 4 个工程实践案例分析以及 10 个工程实训项目,每个案例包括实训目的、实训要求、需求分析、总体设计、详细设计以及实训小结。所有程序都在 Visual C++ 6.0 开发环境中进行了严格的测试,在作者教学网站上也提供了相应的教学支持内容。

本书可作为《C 语言程序设计》课程实验、课程设计以及课程实训的教材,也可供其他从事软件开发工作的读者参考使用。本书不但适合高等学校学生使用,而且也适合初学程序设计者或有一定编程实践基础、希望突破编程难点的读者作为自学教材使用。

## 图书在版编目(CIP)数据

C 语言程序设计实践教程 / 韩立毛,徐秀芳,董琴编

著. — 南京:南京大学出版社,2014.8

普通高校本科计算机类“十二五”规划教材

ISBN 978-7-305-13553-8

I. ①C… II. ①韩… ②徐… ③董… III. ①C 语言—  
程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 153767 号

出版发行 南京大学出版社

社 址 南京市汉口路 22 号 邮 编 210093

出版人 金鑫荣

丛 书 名 普通高校本科计算机类“十二五”规划教材

书 名 C 语言程序设计实践教程

编 著 韩立毛 徐秀芳 董 琴

责任编辑 吴宜错 何永国 编辑热线 025-83397482

照 排 南京南琳图文制作有限公司

印 刷 宜兴市盛世文化印刷有限公司

开 本 787×1092 1/16 印张 15.25 字数 371 千

版 次 2014 年 8 月第 1 版 2014 年 8 月第 1 次印刷

印 数 3000

ISBN 978-7-305-13553-8

定 价 28.00

网址: <http://www.njupco.com>

新浪微博: <http://weibo.com/njupco>

官方微信号: njupress

销售咨询热线: (025) 83594756

\* 版权所有,侵权必究

\* 凡购买南大版图书,如有印装质量问题,请与所购  
图书销售部门联系调换

## 前 言

随着计算机技术的飞速发展,社会对大学生的计算机应用与软件开发水平的要求也在不断提高。程序设计是计算机应用与软件开发的基础,如果只会简单的计算机操作,但不了解软件开发的实质,就无法从根本上理解计算机的工作原理,也很难满足现代社会不断发展的需求。因此,掌握程序设计解决工程实践问题已成为对大学生计算机应用能力的要求之一。

C语言作为一种通用的计算机程序设计语言,其结构简单、数据类型丰富、运算灵活方便,是一种理想的结构化程序设计语言。用C语言可编写高效简洁、风格优美的计算机应用程序和系统程序。用C语言编写的程序,具有运算速度快、效率高、目标代码紧凑、可移植性好等特点。由于C语言功能强大、使用灵活,对于初学程序设计的人来说学习起来比较困难,且很难用于工程实践解决实际问题。鉴于这种情况,为了使初学程序设计的读者能够快速掌握C语言程序设计方法,并初步具备使用C语言开发应用程序和解决实际问题的能力,我们根据多年的教学实践和经验,确定了本书的结构体系,精心选择和编写了课程实践内容。教材内容的编写,充分体现了工程实践应用能力的训练,将理论基础知识运用于工程实践。

本书作为《C语言程序设计》课程的配套教材,结合C语言程序设计课程实践性很强的特点以及初学者学习时的难点,构建了实验操作、模拟实战、工程实训为一体的实践教学体系,通过该实践教学体系来培养读者程序设计的逻辑思维,提高读者程序设计的能力,同时要让读者深刻地领会如何运用软件工程的方法去开发软件工程项目,真正达到学以致用效果。

由于C语言程序设计课程是面向学校不同专业开设的,对不同专业学生的实践能力要求应有所不同。为此,在教材的内容编写时已经充分考虑,授课教师可根据所任专业的要求进行选择。全书共分为3个部分。第1部分为实验操作,共包含16个实验,每个实验包括实验目的、相关知识点、实验内容、具体要求以及程序设计提示。这一部分可根据专业人才培养方案教学计划的学时数选择实验内容,对于教学计划为16学时的实验,可选择其中的1.1、1.2、1.3、1.5、1.7、1.9、1.12、1.16;对于教学计划为24学时的实验,可选择其中的1.1、1.2、1.3、1.5、1.7、1.9、1.10、1.12、1.13、1.14、1.15、1.16;对于教学计划为32学时的实验,可选择其中的1.1至1.16。每个实验要求完成前4道题,其余的题可作为学有余力的学生进行练习。第2部分为模拟实战,共包含20套模拟实战题,每套模拟实战题包括完

善程序、程序调试以及程序设计。第 3 部分为工程实训,共包含 4 个工程实践案例分析以及 10 个工程实训项目。每个案例包括实训目的、实训要求、需求分析、总体设计、详细设计以及实训小结。工程实训项目难易程度不同,读者可参考工程实训案例分析完成这些工程项目,简单的项目可以一个人单独完成,复杂的项目可由几个人共同完成,读者应在完成基本任务的前提下,对程序加以改进和提高。教材中所有程序都在 Visual C++6.0 开发环境中进行了严格的测试,在作者教学网站上也提供了相应的教学支持内容。

本书的作者长期从事 C 语言程序设计课程的理论教学与实践指导工作,并曾利用 C 语言开发过多个软件工程项目,有着丰富的教学与实践经验。本着加强基础、注重实践、勇于创新、突出应用的原则,力求使本教材达到可读性、适用性和先进性,适合应用型本科高校相关专业《C 语言程序设计》课程的实践教学。为了便于读者自学,在教材的体系结构和内容上注意由浅入深、深入浅出、循序渐进的方针;为了提高读者的编程技巧,教材中包含了一些典型范例,不仅适用于教学,同时也适用于用 C 语言开发应用程序的用户参考。

全书由韩立毛、徐秀芳、董琴共同编写,其中韩立毛主要负责第 1 部分的编写,董琴主要负责第 2 部分的编写,徐秀芳主要负责第 3 部分的编写,并由韩立毛负责全书的总体策划与统稿、定稿工作。本书在编写过程中得到了江苏省应用型本科高校计算机系列教材编委会的支持,得到了盐城工学院教材出版基金的资助,同时也得到了本校赵雪梅、刘其明等老师和南京大学出版社吴宜锴编辑的热心帮助和大力支持,在此一并表示感谢。

由于作者水平有限,书中难免存在错误和不足之处,恳请读者批评指正。

编者

2014 年 2 月

# 目 录

<b>第 1 部分 实验操作篇</b> .....	1
1.1 C 程序设计上机初步 .....	1
1.2 顺序结构程序设计 .....	6
1.3 选择结构程序设计.....	14
1.4 循环结构程序设计 1 .....	22
1.5 循环结构程序设计 2 .....	26
1.6 数组及其应用 1 .....	31
1.7 数组及其应用 2 .....	37
1.8 函数与模块化程序设计 1 .....	42
1.9 函数与模块化程序设计 2 .....	49
1.10 编译预处理及程序调试 .....	55
1.11 指针及其应用 1 .....	59
1.12 指针及其应用 2 .....	66
1.13 字符串及其应用 .....	72
1.14 结构体与共用体及其应用 .....	78
1.15 链表及其应用 .....	87
1.16 文件及其应用.....	101
<b>第 2 部分 模拟实战篇</b> .....	113
2.1 上机模拟题 1 .....	113
2.2 上机模拟题 2 .....	115
2.3 上机模拟题 3 .....	118
2.4 上机模拟题 4 .....	121
2.5 上机模拟题 5 .....	123
2.6 上机模拟题 6 .....	126
2.7 上机模拟题 7 .....	129

2.8	上机模拟题 8	133
2.9	上机模拟题 9	136
2.10	上机模拟题 10	139
2.11	上机模拟题 11	142
2.12	上机模拟题 12	144
2.13	上机模拟题 13	148
2.14	上机模拟题 14	151
2.15	上机模拟题 15	154
2.16	上机模拟题 16	157
2.17	上机模拟题 17	160
2.18	上机模拟题 18	163
2.19	上机模拟题 19	166
2.20	上机模拟题 20	169
<b>第 3 部分 工程实训篇</b>		173
3.1	工程实训指导	173
3.2	学生成绩管理系统案例	176
3.3	通讯录管理系统案例	197
3.4	工资管理系统案例	208
3.5	俄罗斯方块游戏系统案例	224
3.6	软件工程实训项目	233
<b>参考文献</b>		238

# 第 1 部分 实验操作篇

## 1.1 C 程序设计上机初步

### 1.1.1 实验目的

1. 熟悉 C 语言程序的运行环境,了解所用计算机系统的软件和硬件配置。
2. 掌握在 Visual C++6.0 环境下编辑、编译、连接和运行 C 语言程序的方法。
3. 通过简单 C 语言程序的操作,掌握 C 语言程序的基本结构及特点。

### 1.1.2 相关知识

#### 1. C 程序的组成

一个完整的 C 语言程序是由一个 main() 函数和若干个其他函数组合而成的,或仅由一个 main() 函数构成。每个完整的 C 程序都必须有且仅有一个 main 函数,程序总是从 main 函数开始执行,而 main 函数可以位于源程序文件中的任何位置。main 函数是程序执行的入口,其他函数的执行是由 main 函数中的语句调用来完成的。被调函数既可以是由系统提供的库函数,也可以是由设计人员自己根据需要而设计的函数。

#### 2. C 函数的结构

一个函数由函数首部和函数体两部分组成,其一般格式如下:

[函数类型] 函数名([函数形式参数表])

```
{  
    数据说明部分;  
    函数执行部分;  
}
```

(1) 函数首部,即函数的第一行。包括函数返回值类型、函数名、函数属性、形式参数类型、形式参数名。

一个函数名后面必须跟一对圆括号,括号内写函数的参数类型及参数名。如果函数没有参数,可以在括号中写 void,也可以是空括号,如 main(void) 或 main()。

(2) 函数体,即函数首部下面的大括弧{……}内的部分。如果一个函数内有多对大括弧,则最外层的一对{}为函数体的范围。

函数体一般包括数据说明和函数执行两个部分。数据说明部分:由变量定义、自定义函数声明和外部变量说明等部分组成,其中变量定义是主要的。函数执行部分:一般由若干条可执行语句组成。

#### 3. C 程序的上机过程

一个 C 语言程序的上机过程一般为:编辑→编译→连接→执行。

(1) 编辑。使用一个文本编辑器编辑 C 语言源程序,并将其保存为文件扩展名为“.c”



的文件。

(2) 编译。将编辑好的 C 语言源程序翻译成二进制目标代码的过程。编译时首先检查源程序的每一条语句是否有语法错误,当发现错误时,就在屏幕上显示错误的位置和错误类型信息,此时要再次调用编辑器进行查错并修改,然后再进行编译,直到排除所有的语法和语义错误。正确的源程序文件经过编译后,在磁盘上生成同名的目标文件(.obj)。

(3) 连接。将目标文件和库函数等连接在一起形成一个扩展名为“.exe”的可执行文件。如果函数名称写错或漏写包含库函数的头文件,则可能提示错误信息,从而得到程序错误数据。

(4) 执行。可以脱离 C 语言编译系统,直接在操作系统下运行。若执行程序后达到预期的目的,则 C 程序的开发工作到此完成,否则要进一步修改源程序,重复编辑—编译—连接—运行的过程,直到取得正确结果为止。

#### 4. C 程序的上机步骤

(1) 源程序编辑和保存。

##### ① 新建源程序。

在 VC++ 开发环境主窗口选择 File(文件)|New(新建)命令,弹出一个 New(新建)对话框,如图 1-1 所示。单击对话框上方的 Files(文件)选项,在其左侧列表中选择 C++ Source File 项,然后分别在右侧的文本框 FileName(文件名)和 Location(位置)中输入准备编辑的源程序文件名和存储路径(如:C:\C 语言\ex1\_1.c)。

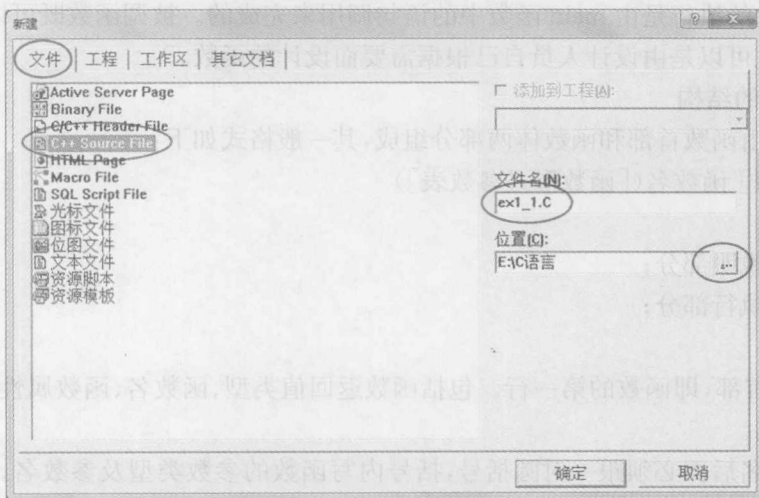


图 1-1 New(新建)对话框

**注意:**文件的扩展名一定要用“.c”,以确保系统将输入的源程序文件作为 C 文件保存。否则,系统默认为 C++ 源文件(默认扩展名为 cpp)。

在开发环境右侧的编辑区输入相关程序代码并保存。

##### ② 打开已存在文件。

在 VC++ 开发环境主窗口选择 File|Open 命令,或按 Ctrl+O 键,此时可通过打开文件对话框选择要装入的文件名;也可以直接在“我的电脑”中按路径找到已有的 C 程序名(如:ex1\_1.c),双击此文件名,则进入 VC++ 集成环境,并打开此文件,编辑区中显示相应

的程序可供修改。

Open 命令可打开多种文件,包括源文件、头文件、各种资源文件、工程文件等,并打开相应的编辑器,使文件内容在工作区显示出来,以供编辑修改。

### ③ 保存文件

在 VC++ 开发环境主窗口选择 File(文件)|Save(保存)命令,将修改后的程序保存在原来的文件中。

#### (2) 源程序的编译。

程序全部或部分编写完成后需进行编译才能运行。程序编译后一般可能会出现一些语法错误,需要根据 Output 窗口中的提示信息对程序进行重新修改,直到编译后不再出现错误为止。

单击 Build(组建)菜单,在其下拉菜单中选择 Compile ex1\_1.c(编译 ex1\_1.c),对程序进行编译。根据提示建立一个默认的工程工作区,选“是”按钮确认;紧接着,系统将会问是否要保存当前的 C 文件,回答“是”;然后,系统开始编译当前程序。如果程序正确,即程序中不存在语法错误,则 VC 窗口出现“ex1\_1.obj - 0 error(s), 0 warning(s)”的提示信息,并生成扩展名为“.obj”的目标文件。

#### (3) 程序的连接。

选择 Build(组建)菜单下的 Build(组建)命令,即可进行连接操作,信息窗口显示连接相关信息,若出现错误,按照错误信息修改源文件后重新编译、组建直到生成 exe 可执行文件。

以上介绍的分别是对源程序进行编译与连接,也可以直接用 Build 菜单下的 Build(或按 F7 键)一次性完成编辑与连接。但对初学者来说,还是提倡分步进行程序的编译和连接,因为刚开始学习所编写的程序出错的机会较多,最好等上一步完成正确后再进行下一步操作。对于有经验的程序员来说,在对程序比较有把握时,可以一步完成编译与连接。

#### (4) 程序的运行。

当程序编译、连接均提示无错误信息(0 error(s))后,选择 Build 菜单下的“执行 ex1\_1.exe”命令,或使用相应的功能键 Ctrl+F5,程序开始运行,然后显示程序的输出结果。输出结果的屏幕将等待用户按下任意键后,才返回编辑状态,一个 C 程序的执行过程结束。

#### (5) 关闭工作空间。

选择 File(文件)中的 Close Workspace(关闭工作空间)命令,在弹出的对话框中单击“是”按钮,退出当前程序。

**注意:**调试完程序后重新编写新的程序,一定要用 File|Close Workspace 命令关闭工程文件,否则编译或运行时总是原来的程序。

## 1.1.3 实验内容

### 1. 阅读并运行下列程序

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



```
return z;
```

**【要求】**

- (1) 输入给定的源程序,以 ex01\_3.c 为文件名保存。
- (2) 编译并运行,在运行时从键盘输入 2 和 5,然后按“回车”键,观察运行结果。
- (3) 将程序中的第 5 行改为: int a,b,c;
- 先分析可能出现的情况,再进行编译调试。
- (4) 将 max 函数中的第 5、6 两行合并写为一行,即:

```
if(x>y) z=x;
else z=y;
```

写成:

```
if(x>y) z=x; else z=y;
```

再进行编译和运行,分析运行结果。

- (5) 修改给定的程序使其能够求 a,b,c 中的最大数。
- (6) 用下列数据测试:

第 1 组数据:a,b,c 的值为 2,3,4。

第 2 组数据:a,b,c 的值为 2,4,4。

第 3 组数据:a,b,c 的值为 4,3,2。

**【程序设计提示】**

只要修改 main() 函数即可实现:

```
void main()
{
    int max(int x,int y);
    ..... //定义 a,b,c,d 为整型
    ..... //输入三个整数 a,b,c
    ..... //先求出 a,b 中的较大数,并放入 d 中
    ..... //再求出 c,d 中的较大数
    ..... //输出 d
}
```

**4. 输入并运行下列程序**

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a,b,c,p,s;
    printf("请输入 a,b,c 的值:");
    scanf("%f,%f,%f",&a,&b,&c);
    p=1.0/2*(a+b+c);
    s=sqrt(p*(p-a)*(p-b)*(p-c));
```

```
printf("三角形的面积为:%f\n",s);  
}
```

### 【要求】

- (1) 阅读分析该程序的功能和运行结果。
- (2) 输入给定的源程序,并以 ex01\_4.c 为文件名保存。
- (3) 上机运行该程序,与自己分析的结果比较。
- (4) 将语句: $p=1.0/2 * (a+b+c)$ ; 修改为: $p=1/2 * (a+b+c)$ ;再进行编译和运行,分析运行结果的正确性。
- (5) 用下列数据测试:  
第 1 组数据:a,b,c 的值为 2,3,4。  
第 2 组数据:a,b,c 的值为 3,4,3。

## 1.2 顺序结构程序设计

### 1.2.1 实验目的

1. 掌握 C 语言数据类型的概念,熟悉整型、字符型和实型变量的定义方法,并掌握对它们赋值的方法。
2. 掌握不同数据类型之间赋值的规律。
3. 学会使用 C 语言的算术运算符,以及包含这些运算符的表达式,特别是自加和自减运算符的使用。
4. 掌握各种类型数据的输入输出的方法,能正确使用各种格式转换符。
5. 理解 C 语言程序的顺序结构,掌握简单程序的设计方法。
6. 进一步熟悉 C 程序的编辑、编译、连接和运行的过程。

### 1.2.2 相关知识

#### 1. 标识符

标识符一般用来标识变量名、符号常量名、自定义函数名、数组名、自定义类型名、文件名的有效字符序列。

标识符的命名规则:

- (1) 标识符以字母或下划线开头,只能由字母、数字或下划线组成的字符序列(其实 C99 中有更宽泛的规定)组成。
- (2) 标识符的长度因系统而异,C89 允许的最长标识符为 31 个有效字符,C99 增加到 63 个有效字符。
- (3) 不能将 C 语言的关键字作为标识符。

#### 2. 数据类型

数据类型是用来描述数据的表示形式,用来定义常量或变量允许具有何种形式的数值或对其进行何种操作。在程序设计语言中,通过数据类型描述程序中的数据。

C 语言中的数据类型非常丰富,主要包括四类:基本数据类型、构造类型、指针类型、空值类型。丰富的数据类型使得 C 语言具有很强的数据处理功能。

C语言有三种基本数据类型:整型、实型、字符型。

### 3. 常量

#### (1) 数字常量。

数字常量主要包括普通数字、指数形式、长整型。

普通数字:1,35,2.7

指数形式:2.45e-2 等价于  $2.45 \times 10^{-2}$

**注意:**e 大小写皆可。e 前面的数字不能省,就算是 1 也不能省。e 后面的数字一定要是整数。

长整型,单精度浮点型:3235L、32.5F 分别表示 3235 是长整型数据,32.5 是单精度浮点型,若不写上 L 和 F,则表示 3235 是整型,32.5 是双精度浮点型,L 和 F 大小写皆可。

#### (2) 字符常量。

字符常量主要包括普通字符常量和转义字符常量。

普通字符常量:用单引号把一个字符括起来,如 'A'、'@'。

转义字符常量:一对单引号括起来并以“\”开头的字符序列,如 '\n' (回车)、'\123' (八进制 123 对应的字符)、'\x23' (十六进制 23 对应的字符)。

#### (3) 字符串常量。

用一对双引号把一个字符序列括起来,如"ABCef"。系统存放字符串常量,每个字符分配一个字节,各字符所占字节紧邻,并且字符串末尾会再开辟一个字节存储空间,以存放'\0'作为字符串结束标志。

#### (4) 符号常量。

符号常量定义格式:#define 符号常量名 符号 常量值。如#define N 20 定义了符号常量 N,其值为 20。注意符号常量名和符号常量值之间是用空格隔开,而不是用等号分开。#define 和符号常量名之间也是用空格隔开。

### 4. 变量

#### (1) 变量概念。

变量是指在程序运行过程中其值可以改变的量。程序中用到的所有变量都必须有一个名字作为标识。变量的名字由用户定义,它必须符合标识符的命名规则。

#### (2) 变量定义的一般格式。

[存储类型] 数据类型 变量名 1,变量名 2,...;

例如:

```
int a,b,c;           //定义 3 个整型变量 a,b,c
long m,n;           //定义 2 个长整型变量 m,n
float x,y;          //定义 2 个实型变量 x,y
char c1,c2;         //定义 2 个字符型变量 c1,c2
```

#### (3) 自增自减运算。

“变量++，++变量，变量--，--变量”：使变量的值自增 1 或自减 1，等价于：变量=变量+1，变量=变量-1。

“++，--”放于变量前后效果的区别：

当自增自减运算作为表达式的一部分时，“++，--”放在变量前面是先自增、自减再

使用变量的值,放在变量后面则是先使用变量的值,再自增、自减。

例如:

```
x=3; printf("%d", ++x);
```

相当于执行了“++x; printf("%d",x);”的操作,所以输出结果为 4。

再例如:

```
x=3; printf("%d", x++);
```

相当于执行了“printf("%d",x); x++;”的操作,输出结果为 3,当然最后 x 的值还是 4。

## 5. 运算符与表达式

### (1) C 语言的运算符。

C 语言的运算符非常丰富,见表 1-1 所示。

表 1-1 C 语言运算符

名称	功能	符号
算术运算符	用于各类数值运算	+ - * / % ++ --
关系运算符	用于比较运算	> < == >= <= !=
逻辑运算符	用于逻辑运算	&&    !
赋值运算符	用于赋值运算	= += -= *= /= %= <<= >>= &=  =
条件运算符	三目运算符	?:
位操作运算符	按二进制位进行运算	&   ~ · << >>
逗号运算符		,
求字节数运算符		sizeof
指针运算符		* &
特殊运算符		() [] · ->

### (2) 算术表达式。

用算术运算符和括号将运算对象(也称操作数)连接起来的、符合 C 语法规则的等式,称为 C 算术表达式。运算对象包括常量、变量、函数等。算术运算符的优先级见表 1-2 所示。

表 1-2 算术运算符的优先级

运算符	说明	优先级
( )	圆括号	高 ↓ 低
-、++、--	单目运算符,取负、自加、自减	
*、/、%	双目运算符,乘、除、取余	
+、-	双目运算符,加、减	

**注意:**当优先级相同的运算符同时出现在表达式中时,算术运算符的结合方向为“自左向右”,即从左向右依次计算。如  $9-3+4$  的运算顺序为先计算  $9-3$  结果为 6,再计算  $6+4$  结果为 10。

两种类型转换。一种是在运算时不必用户指定,系统自动进行的类型转换,如 $3+6.5$ 。第二种是强制类型转换。当自动类型转换不能达到目的时,可以用强制类型转换运算符将一个表达式转换成所需类型。

**注意:**++、--运算,++i、--i是在使用i之前,先使i的值加(减)1;i++、i--是在使用i之后,使i的值加(减)1。++和--的结合方向是“自右至左”,只能用于变量,而不能用于常量或表达式。

## 6. 赋值类型转换

(1) 将浮点型数据赋给整型变量时,舍弃浮点数的小数部分。

(2) 将整型数据赋给单、双精度变量时,数值不变,但以浮点数形式存储到变量中(注意长度,如当f为float变量时,先执行 $f=2$ ,再将2转换为2.000000,最后存储到f中)。

(3) 将一个double型数据赋给float变量时,截取其前面7位有效数字,但应注意数值范围不能溢出。

(4) 字符型数据直接赋给整型变量。

(5) 将一个int、short、long型数据赋给一个char型变量时,只将其低于8位的数值原封不动地送到char型变量(即截断)。反之,若将一个long型数据赋给一个int型变量,将long型数据中低于16位的数值原封不动地送到整型变量。

## 7. 数据的输入输出

C语言本身不提供输入输出语句,输入和输出操作是由C函数库中的函数来实现的。在使用系统库函数时,要用预编译命令#include "stdio.h"或#include <stdio.h>。常用的有下面几种输入输出函数:

(1) 字符输出函数: putchar, 格式: putchar(c), 功能: 输出一个字符。

(2) 字符输入函数: getchar, 格式: getchar(), 功能: 输入一个字符。

(3) 格式输出函数: printf, 格式: printf(格式控制, 输出表列), 功能: 输出若干个任意类型的数据。

格式控制主要是输出格式和转义字符。

格式控制字符包括格式转换说明符和普通字符两部分。格式控制说明符表示按指定的格式输出数据,普通字符是在输出数据时需要输出原有字符。格式控制符见表1-3所示。

表1-3 格式控制符

%d	以带符号的十进制形式输出整数
%o	以八进制无符号形式输出整数
%x	以十六进制无符号形式输出整数
%u	以无符号十进制形式输出整数
%c	以字符形式输出,只输出一个字符
%s	输出字符串
%f	以小数形式输出单、双精度数,隐含输出六位小数
%e	以指数形式输出实数
%g	选用%f或%e格式中输出宽度较短的一种格式,不输出无意义的0



格式输出的四个常用附加格式说明字符:

i 或 L 用于长整型整数; m 是数据的最小宽度; n 对于实数, 表示输出 n 位小数; 而对于字符串, 则表示截取的字符个数; -(减号) 输出的数字或字符在域内向左靠。

(4) 格式输入函数格式: scanf(格式控制, 地址表列), 功能: 按照变量在内存的地址将变量值存进去。

例如: int a, b, c;

```
scanf("%d%d%d",&a,&b,&c); //这里"&."是取地址
```

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

格式输入的四个常用附加格式说明字符:

l 用于输入长整型数据; h 用于输入短整型数据; \* 用于表示本输入项在读入后不赋给相应的变量。域宽 m 用来指定输入数据所占宽度(列数)。

## 8. 顺序结构程序设计

### (1) 问题分析。

此类问题的解决是用顺序结构按照编写代码的顺序依次执行相关计算或处理。分析: ① 实现要完成功能的方法步骤; ② 输入数据及其类型; ③ 对输入数据的处理; ④ 输出数据及其格式。

### (2) 算法分析。

此类问题的算法一般都很简单, 主要是对一些初始数据的计算或处理。

### (3) 代码设计。

① 用 scanf 函数输入原始数据;

② 用赋值语句进行计算或处理;

③ 用 printf 函数输出计算或处理的结果。

### (4) 运行调试。

用初始数据的不同情况分别测试程序和运行结果。

## 1.2.3 实验内容

### 1. 阅读并运行下列程序

```
#include <stdio.h>

void main()
{
    char c1,c2;
    c1='a';
    c2='b';
    printf("%c,%c\n",c1,c2);
}
```

#### 【要求】

(1) 输入给定的源程序, 并以 ex02\_1.c 为文件名保存。

(2) 编译并运行此程序, 分析程序的运行结果。

(3) 在 printf 语句的下面再增加一条语句: printf("%d,%d\n",c1,c2); 再进行编译与