

高等学校计算机应用规划教材

C 语言程序设计

学习指导与实验教程

(第五版)

冯相忠 主 编

潘洪军 亓常松 叶其宏 吴远红 副主编

清华大学出版社

北 京

内 容 简 介

本书是与教材《C 语言程序设计(第五版)》相配套的学习指导与实验用书,内容主要包括 C 语言程序设计学习指导和 C 语言程序设计实验教程两部分。在学习指导部分,介绍 C 语言各章知识的要点和难点,选择了一些典型例题进行分析,选编了一些练习题。这些练习题题型丰富、覆盖面广,包括选择题、填空题、阅读程序写结果题、编写程序题,并且每道题都给出了参考答案。通过这些练习题,可以训练读者理解和掌握 C 语言的基本概念与基本语句,学会编写程序以及掌握编程的方法和技巧。在实验教程部分,介绍了 10 个实验内容,为读者在计算机上进行程序的编辑、调试和运行提供了详细的指导。对于每个实验,列出了实验目的、实验知识内容提要、实验的具体内容以及实验完成后的思考问题。通过这些实验,提高读者的实际动手能力。

本书条理清晰、语言流畅、通俗易懂,实用性强,既可作为高等院校应用型本科专业的教材,也可供自学者以及参加 C 语言计算机等级考试的人员阅读使用。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计学习指导与实验教程 / 冯相忠 主编. —5 版. —北京:清华大学出版社, 2020.7
高等学校计算机应用规划教材

ISBN 978-7-302-55847-7

I. ①C… II. ①冯… III. ①C 语言—程序设计—高等学校—教学参考资料 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2020)第 106062 号

责任编辑:胡辰浩

封面设计:孔祥峰

版式设计:妙思品位

责任校对:成凤进

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市国英印务有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 15.25 字 数: 390 千字

版 次: 2011 年 1 月第 1 版 2020 年 8 月第 5 版 印 次: 2020 年 8 月第 1 次印刷

印 数: 1~3000

定 价: 62.00 元

产品编号: 088808-01

前 言

C 语言是被广泛使用的一种计算机语言，由于它具有功能丰富、灵活性强、可移植性好、语言简洁、应用面广等特性，因此深受广大用户的喜爱。

对于 C 语言，初学者可能感觉学起来比较吃力，具体表现为：感觉 C 语言的语法难以理解，C 语言的语句不易正确使用，设计和编写程序时不知如何下手，综合运用 C 语言知识处理实际问题较棘手。有鉴于此，为了帮助读者学好 C 语言，顺利通过各级各类相关考试，并能熟练将其运用于实际工作中，我们组织长期从事 C 语言教学工作的教师，编写了这本《C 语言程序设计学习指导与实训教程(第五版)》。

本书内容分为两部分，一部分是 C 语言的学习指导；另一部分是 C 语言的实训教程。

在学习指导部分，对 C 语言各章知识的要点和重点进行了整理归纳和深入分析，以使读者明确各章应该着重掌握的知识和深入理解的问题。在学习指导部分，选择了一些典型例题进行分析，以使读者理解并能够灵活运用各章知识。在学习指导部分，选编了相应习题，每道习题基本上都给出了参考答案。这些习题既可以训练读者理解和掌握 C 语言的基本概念与基本语句，又可以训练读者学习编写程序的方法和技巧。对于每一道编写程序题，本书只给出了一种参考答案，因为每个问题一般都有多种编程方法，所以读者不要受参考答案的限制。

在实训教程部分，根据 C 语言的教学内容，本书设计了 10 个上机练习实训，为读者在计算机上进行程序的编辑、运行和调试提供详细的指导。对于每个实训，均列出了该实训要达到的目的、所涉及的 C 语言知识要点、具体操作以及实训完成后的思考问题。认真完成这些实训，读者将能够极大地提高编程能力。

封面署名的浙江海洋大学的主编和副主编外，参与本书编写的还有王广伟、乐天、毕振波、江有福、朱本浩、宋广军、张建科、荣品、侯志凌、管林挺、谭小球、谭安辉等。

由于编者水平有限，书中难免存在错误与不足之处，诚恳欢迎读者批评指正。我们的邮箱是 huchenhao@263.net，电话是 010-62796045。

编 者
2020 年 4 月

目 录

第 1 篇 C 语言程序设计学习指导	
第 1 章 C 语言概述	1
1.1 本章要点	1
1.1.1 C 语言的优点	1
1.1.2 C 语言程序的结构	1
1.1.3 C 语言源程序的编辑、编译、 链接和运行	1
1.2 习题	2
1.3 习题参考答案	2
第 2 章 数据类型、运算符和表达式	3
2.1 本章要点	3
2.1.1 C 语言的数据类型	3
2.1.2 标识符	3
2.1.3 常量	3
2.1.4 变量	4
2.1.5 数据类型的转换	4
2.1.6 C 语言运算符的优先级和结合性	5
2.1.7 算术运算符和算术表达式	6
2.1.8 自增运算符和自减运算符	6
2.1.9 赋值运算符和赋值表达式	6
2.1.10 复合赋值运算符	6
2.1.11 逗号运算符	6
2.2 本章难点	6
2.2.1 运算符的优先级	6
2.2.2 运算符的结合性	7
2.2.3 复合赋值运算符	7
2.2.4 赋值类型转换	7
2.3 例题分析	8

2.4 习题	9
2.4.1 单项选择题	9
2.4.2 填空题	12
2.4.3 阅读程序写结果题	13
2.4.4 编写程序题	15
2.5 习题参考答案	15
2.5.1 单项选择题答案	15
2.5.2 填空题答案	15
2.5.3 阅读程序写结果题答案	15
2.5.4 编写程序题参考答案	15
第 3 章 程序设计初步	17
3.1 本章要点	17
3.1.1 格式输入函数scanf()	17
3.1.2 格式输出函数printf()	18
3.1.3 单字符输入函数getchar()	18
3.1.4 单字符输出函数putchar()	19
3.1.5 赋值语句	19
3.1.6 复合语句和空语句	19
3.1.7 顺序结构	19
3.2 本章难点	20
3.2.1 printf()函数中的格式字符、附加 格式符和转义字符	20
3.2.2 scanf()函数中的格式字符和附加 格式符	20
3.3 例题分析	21
3.4 习题	22
3.4.1 单项选择题	22
3.4.2 填空题	24
3.4.3 阅读程序写结果题	25

3.4.4 编写程序题	26	5.2.2 3种循环结构的比较	50
3.5 习题参考答案	27	5.2.3 多重循环	50
3.5.1 单项选择题答案	27	5.3 例题分析	51
3.5.2 填空题答案	27	5.4 习题	54
3.5.3 阅读程序写结果题答案	27	5.4.1 单项选择题	54
3.5.4 编写程序题参考答案	27	5.4.2 填空题	55
第4章 选择结构程序的设计	31	5.4.3 阅读程序写结果题	57
4.1 本章要点	31	5.4.4 编写程序题	61
4.1.1 关系运算符与关系表达式	31	5.5 习题参考答案	62
4.1.2 逻辑运算符与逻辑表达式	32	5.5.1 单项选择题答案	62
4.1.3 二分支选择结构——if语句	32	5.5.2 填空题答案	62
4.1.4 条件运算符和条件表达式	33	5.5.3 阅读程序写结果题答案	62
4.1.5 多分支选择结构——switch语句	34	5.5.4 编写程序题参考答案	62
4.2 本章难点	34	第6章 数组	69
4.2.1 if语句的嵌套	34	6.1 本章要点	69
4.2.2 条件表达式的使用	35	6.1.1 一维数组	69
4.2.3 switch语句和break语句的使用	35	6.1.2 二维数组	69
4.3 例题分析	35	6.1.3 字符数组	70
4.4 习题	39	6.2 本章难点	71
4.4.1 单项选择题	39	6.2.1 数组元素的下标	71
4.4.2 填空题	41	6.2.2 字符串和字符数组	71
4.4.3 阅读程序写结果题	42	6.2.3 字符串处理函数	71
4.4.4 编写程序题	43	6.3 例题分析	72
4.5 习题参考答案	44	6.4 习题	74
4.5.1 单项选择题答案	44	6.4.1 单项选择题	74
4.5.2 填空题答案	44	6.4.2 填空题	76
4.5.3 阅读程序写结果题答案	44	6.4.3 阅读程序写结果题	77
4.5.4 编写程序题参考答案	44	6.4.4 编写程序题	81
第5章 循环结构程序的设计	47	6.5 习题参考答案	82
5.1 本章要点	47	6.5.1 单项选择题答案	82
5.1.1 while语句构成的循环	47	6.5.2 填空题答案	82
5.1.2 do-while语句构成的循环	47	6.5.3 阅读程序写结果题答案	83
5.1.3 for语句构成的循环	48	6.5.4 编写程序题参考答案	83
5.1.4 goto语句以及goto语句构成的循环	48	第7章 函数	91
5.1.5 多重循环	49	7.1 本章要点	91
5.1.6 break语句和continue语句	49	7.1.1 函数的定义	91
5.2 本章难点	50	7.1.2 函数的参数和返回值	92
5.2.1 循环结构的理解	50	7.1.3 函数调用	92

7.1.4 函数声明	93	8.5 习题参考答案	117
7.1.5 数组名作为函数参数	93	8.5.1 单项选择题答案	117
7.1.6 全局变量和局部变量	93	8.5.2 填空题答案	117
7.1.7 变量的存储类别	93	8.5.3 阅读程序写结果题答案	117
7.1.8 内部函数与外部函数	94	8.5.4 编写程序题参考答案	117
7.2 本章难点	94	第 9 章 指针	121
7.2.1 参数的传递	94	9.1 本章要点	121
7.2.2 函数的声明	94	9.1.1 指针变量的定义	121
7.2.3 函数的递归调用	94	9.1.2 指针变量的赋值	121
7.2.4 数组名作为函数参数	95	9.1.3 对指针变量的操作	122
7.2.5 变量的作用域	95	9.1.4 指向指针变量的指针变量	123
7.2.6 静态存储类别	95	9.1.5 数组与指针	123
7.3 例题分析	95	9.1.6 指针数组	124
7.4 习题	97	9.1.7 指向函数的指针变量	124
7.4.1 单项选择题	97	9.1.8 带参数的main()函数	124
7.4.2 填空题	99	9.1.9 void类型的指针	125
7.4.3 阅读程序写结果题	102	9.2 本章难点	125
7.4.4 编写程序题	104	9.2.1 指针变量的概念	125
7.5 习题参考答案	104	9.2.2 对指针变量的操作	126
7.5.1 单项选择题答案	104	9.2.3 字符指针变量与字符数组的区别	126
7.5.2 填空题答案	104	9.3 例题分析	127
7.5.3 阅读程序写结果题答案	105	9.4 习题	132
7.5.4 编写程序题参考答案	105	9.4.1 单项选择题	132
第 8 章 预处理命令	109	9.4.2 填空题	135
8.1 本章要点	109	9.4.3 阅读程序写结果题	137
8.1.1 不带参数的宏	109	9.4.4 编写程序题(要求使用指针)	139
8.1.2 带参数的宏	109	9.5 习题参考答案	140
8.1.3 文件包含	110	9.5.1 单项选择题答案	140
8.1.4 条件编译	110	9.5.2 填空题答案	140
8.2 本章难点	111	9.5.3 阅读程序写结果题答案	140
8.2.1 宏展开	111	9.5.4 编写程序题参考答案	140
8.2.2 条件编译	111	第 10 章 结构体与其他数据类型	145
8.3 例题分析	111	10.1 本章要点	145
8.4 习题	112	10.1.1 结构体概述	145
8.4.1 单项选择题	112	10.1.2 定义结构体类型变量的方法	145
8.4.2 填空题	115	10.1.3 结构体变量的引用和初始化	145
8.4.3 阅读程序写结果题	116	10.1.4 结构体数组	146
8.4.4 编写程序题	117	10.1.5 指向结构体数据的指针	146

10.1.6	用指针处理链表	147
10.1.7	共用体	148
10.1.8	枚举类型	148
10.1.9	用typedef定义类型	148
10.2	本章难点	149
10.2.1	在函数之间传递结构体数据	149
10.2.2	结构体与共用体的区别	149
10.2.3	链表操作	150
10.3	例题分析	150
10.4	习题	153
10.4.1	单项选择题	153
10.4.2	填空题	156
10.4.3	阅读程序写结果题	160
10.4.4	编写程序题	163
10.5	习题参考答案	163
10.5.1	单项选择题答案	163
10.5.2	填空题答案	164
10.5.3	阅读程序写结果题答案	164
10.5.4	编写程序题参考答案	164
第 11 章	位运算	169
11.1	本章要点	169
11.1.1	位运算符和位运算	169
11.1.2	位运算的优先级	170
11.1.3	位段	171
11.2	例题分析	171
11.3	习题	173
11.3.1	单项选择题	173
11.3.2	填空题	173
11.3.3	阅读程序写结果题	174
11.3.4	编写程序题	175
11.4	习题参考答案	175
11.4.1	单项选择题答案	175
11.4.2	填空题答案	175
11.4.3	阅读程序写结果题答案	175
11.4.4	编写程序题参考答案	176

第 12 章	文件	177
12.1	本章要点	177
12.1.1	文件概述	177
12.1.2	文件类型指针	177
12.1.3	文件的打开和关闭	177
12.1.4	文件的读/写	179
12.1.5	文件定位	180
12.1.6	文件检测	181
12.2	本章难点	181
12.2.1	文件位置指针的合理定位	181
12.2.2	各文件读/写函数的区别	182
12.3	例题分析	182
12.4	习题	185
12.4.1	单项选择题	185
12.4.2	填空题	188
12.4.3	阅读程序写结果题	189
12.4.4	编写程序题	190
12.5	习题参考答案	191
12.5.1	单项选择题答案	191
12.5.2	填空题答案	191
12.5.3	阅读程序写结果题答案	191
12.5.4	编写程序题参考答案	191

第 2 篇 C 语言程序设计实验教程

实验一	数据类型、运算符和表达式 ..	195
实验二	简单C程序设计	199
实验三	选择结构程序的设计	203
实验四	循环结构程序的设计	207
实验五	数组的应用(一) ——数值数据的数组	211
实验六	数组的应用(二) ——字符数据的数组	215
实验七	函数	219
实验八	指针	223
实验九	结构体	227
实验十	文件	231

第1篇 C语言程序设计学习指导

第 1 章

C语言概述

1.1 本章要点

1.1.1 C 语言的优点

- (1) 语言简洁、紧凑，使用方便、灵活，运算符和数据结构丰富。
- (2) 允许直接访问物理地址，能够进行位操作，可实现汇编语言的大部分功能，可以直接对硬件进行操作。
- (3) 具有结构化的控制语句，是结构化的理想语言。
- (4) 语法限制不太严格，程序设计自由度大。
- (5) 编写的程序可移植性好。
- (6) 生成的目标代码质量好，程序执行效率高。

1.1.2 C 语言程序的结构

- (1) 一个源程序由若干个文件组成，每个文件由若干个函数组成，其中有且仅有一个主函数(main 函数)。
- (2) 一个函数由函数首部(函数第一行)和函数体(函数首部后面大括号内的部分)组成。函数首部包括函数类型、函数名和括号中的若干参数；函数体由声明部分和执行部分组成。
- (3) 程序书写格式自由，一行内可写多条语句，一条语句也可以分写在多行上，且语句中的空格和回车符均可忽略不计。
- (4) 程序的注释内容放在/*和*/之间，/和*之间不允许有空格；注释部分允许出现在程序中的任何位置。

1.1.3 C 语言源程序的编辑、编译、链接和运行

对于使用 Visual C++、C-Free 或 Turbo C 编写的 C 语言源程序，经过编译、链接生成可执行文件后，即可运行。

1.2 习 题

1. 简述 C 语言源程序的结构特点。
2. 写出 C 语言中一个函数的构成。
3. 编写完成如下任务的程序，然后上机编译、链接并运行。

输出两行字符，第 1 行是 “The computer is our good friends!”，第 2 行是 “We learn C language。”。

4. 编写完成如下任务的程序，然后上机编译、链接并运行。

已知 $a=10$ 、 $b=5$ ，计算 $a+b$ 、 $a-b$ 的值，输出计算结果。

1.3 习题参考答案

1. (略)

2. (略)

3.

```
#include <stdio.h>

main()
{printf("The computer is our good friends!\n");
 printf("We learn C language.\n");
 return 0;
}
```

4.

```
#include <stdio.h>

main()
{int a=10,b=5,c,d;
 c=a+b; d=a-b;
 printf("a+b=%d,a-b=%d \n",c,d);
 return 0;
}
```

数据类型、运算符和表达式

2.1 本章要点

2.1.1 C 语言的数据类型

C 语言提供了丰富的数据类型，通过这些数据类型可以实现如链表、栈等复杂数据结构的编写。程序中的每一个量(包括常量和变量)都属于某一种特定的数据类型。

2.1.2 标识符

所谓标识符就是指在程序中用于标识函数、变量以及常量且符合一定命名规则的字符串。C 语言标识符可以包括一个或多个字符，字符可以是英文字母、数字或下画线，但第一个字符必须是字母或下画线。

特别提醒：C 语言标识符区分字母大小写；标识符命名不宜过长，最好有明确的含义。

2.1.3 常量

常量是在程序运行过程中其值不可被改变的量。常量的类型包括整型常量、实型常量、字符常量、字符串常量和符号常量这 5 类，下面分别予以介绍。

1. 整型常量

整型常量有 3 种形式，分别为十进制整型常量、八进制整型常量和十六进制整型常量。

特别提醒：八进制数以数字 0 开头，十六进制数以 0x 开头，第一位都不是字母 o，以字母 o 开头不是常量的写法，这需要初学者特别注意。

2. 实型常量

实型常量有两种表示形式，分别为小数形式和指数形式。

3. 字符常量

一个字符常量代表 ASCII 字符集中的一个字符，在程序中用单引号(')引起来。

特别提醒：'a'和'A'是两个不同的字符常量；除了这类可打印字符使用的格式以外，对于一些特殊字符常量，采用转义字符的方式表示，要注意转义字符的构成方式。

4. 字符串常量

字符串常量指的是用双引号引起来的一个或多个字符。

5. 符号常量

符号常量是使用宏定义#define 来定义的常量，即用一个标识符代表一个常量，程序中出现该标识符，等同于使用定义时所用到的常量。符号常量可以是上述各常量的任何一种类型。

2.1.4 变量

变量就是其值在程序运行过程中可以改变的量。注意：变量的值可以改变，但并不是非变不可。变量具有变量名，它实际上代表一定的存储单元，其中存储的是该变量的值，通过变量名可以引用其所代表的存储单元中的内容。不同类型的变量其存储单元的大小不同。变量的使用必须符合“先定义，后使用”的原则。变量的类型包括整型、实型、字符型这3类，下面分别予以介绍。

1. 整型变量

整型变量有6种，分别为有符号基本整型(signed int)、无符号基本整型(unsigned int)、有符号短整型(signed short int)、无符号短整型(unsigned short int)、有符号长整型(signed long int)、无符号长整型(unsigned long int)。

2. 实型变量

实型变量包括单精度(float)和双精度(double)两种类型。

3. 字符型变量

字符型变量用于存放一个字符，用关键字 char 来声明。

在C语言中，字符型变量也有带符号与无符号之分，一般情况下，直接使用 char 声明的字符变量是带符号的，其数值范围为-128~127；可以使用 unsigned char 声明无符号字符型变量，其数值范围为0~255。

2.1.5 数据类型的转换

当同一表达式中各个数据的类型不同时，需要把它们转换成同一类型后再进行计算，这种转换可以由编译程序自动实现，即自动转换；也可以由程序员在编译程序时使用类型转换运算符实现，即强制转换。

1. 自动转换

自动转换又可以分为两类，一类是必然实现的转换，即不论参与运算的数据类型是否一致，某些类型的数据也必然转换为另一种类型，主要包括如下两种情况。

- (1) 凡属于 `char`、`short` 类型的变量在运算时一律转换为 `int` 类型。
- (2) 凡属于 `float` 类型的变量在运算时一律转换为 `double` 类型。

另一类是当运算对象的数据类型不同时，按照从低到高的顺序进行转换。例如，若 `a` 是 `int` 类型，`b` 是 `double` 类型，计算 `a/b` 时，将 `a` 转换成 `double` 类型后，再与 `b` 相除。

2. 强制转换

格式为：(数据类型标识)表达式

其作用是把表达式的结果转换为由“数据类型标识”指定的数据类型。例如，`(double)(a+b)`，是将 `a+b` 的值强制转换为 `double` 类型。注意，`(double)(a+b)` 与 `(double)a+b` 不同。

2.1.6 C 语言运算符的优先级和结合性

所谓运算符的优先级，是指当一个表达式中存在两个(或两个以上的)运算符时，根据运算符的优先级决定先执行哪个运算符？例如对于 `a+2×b`，因乘法的优先级高于加法，故先执行乘法，后执行加法。所谓运算符的结合性，是指当一个操作数两侧的运算符具有相同的优先级时，该操作是与左边的运算符结合，还是先与右边的运算符结合？可分为自左至右、自右至左两种结合方向。例如，加减法是自左向右结合的，对于 `a-b+c`，变量 `b` 先与减号结合，执行 `a-b`，然后再执行 `+c` 运算。

本章涉及的具体运算符的优先级和结合性如表 1-2-1 所示。

表 1-2-1 C 语言运算符的优先级和结合性

优 先 级	运 算 符	名 称	结 合 方 向
1	<code>++</code>	增 1 运算符	自右向左
	<code>--</code>	减 1 运算符	
	<code>-</code>	负号运算符	
	<code>(类型)</code>	类型转换运算符	
	<code>sizeof</code>	长度运算符	
2	<code>*</code>	乘法运算符	自左向右
	<code>/</code>	除法运算符	
	<code>%</code>	取模运算符	
3	<code>+</code>	加法运算符	自左向右
	<code>-</code>	减法运算符	
4	<code>=</code> <code>+=</code> <code>-=</code> <code>*=</code>	赋值运算符	自右向左
	<code>/=</code> <code>%=</code>		
5	<code>,</code>	逗号运算符	自左向右

2.1.7 算术运算符和算术表达式

算术运算符包括加(+)、减(-)、乘(*)、除(/)、取模(%),其运算符符合数学上的运算规则。需要说明的是:两个整型量相除时,所得的结果为整型,如 3/2 的结果为 1,而非 1.5;取模运算要求连接的两个量必须是整型数据。

算术表达式指用算术运算符和括号将运算量连接起来、符合 C 语言语法规则的表达式。括号可以改变运算符的自然运算顺序,注意只能使用小括号。

2.1.8 自增运算符和自减运算符

自增运算是使变量的值增 1,而自减运算是使变量的值减 1。

特别提醒:

- (1) 只有变量才能用自增(减)运算符连接,不可以将该类运算符用于表达式或常量。
- (2) 自增(减)运算有前缀方式和后缀方式两种,使用时注意两者的区别。
- (3) 自增(减)运算符为单目运算符,其结合性都是自右向左结合。

2.1.9 赋值运算符和赋值表达式

赋值运算符的一般使用形式为“变量=表达式”,用于连接一个变量(准确一点说是内存单元)与一个表达式,其功能是把右边表达式的值赋予左边的变量。

左边的变量(内存单元)称为左值,右边的表达式称为右值。

2.1.10 复合赋值运算符

在赋值运算符之前加上其他运算符可以构成复合赋值运算符,这主要是为了提高 C 语言编译器的编译效率。例如,与算术运算符结合可以形成+=、-=、*=、/=、%=。

需要说明的是:复合赋值运算符的两个符号之间一定不能有空格,复合赋值运算符与赋值运算符(=)具有相同的优先级和结合性。

2.1.11 逗号运算符

用逗号运算符将两个表达式连接起来所形成的表达式称为逗号表达式,一般形式为:

表达式 1, 表达式 2

需要说明的是:逗号运算符连接的表达式 1 或表达式 2 也可以是逗号表达式。

逗号表达式通常用于连接一些处于并列关系的表达式,可以用在 for 语句中。

2.2 本章难点

2.2.1 运算符的优先级

不同优先级的运算符出现在同一表达式中时,必须根据各运算符的优先级来使用它们,否则运算的结果就可能不正确。

各种类型运算符的优先级有一定的规律(可参照第 2.1 节中关于运算符优先级的内容)可循,如果难以确定各运算符的优先级,可以在表达式中适当地加上一些小括号来避免意外结果的产生。另外,在运算符的两边有意加上若干个空格,也可增强表达式的可读性。

例如,对于表达式 $(M)?(a++):(a--)$,其中的运算符 $++$ 、 $--$ 的优先级相同,且都高于条件运算符 $(?:)$,所以这个表达式也可写成 $M?a++:a--$,其作用与原表达式完全相同,但可读性就远不如原表达式了。

2.2.2 运算符的结合性

运算符的结合性其实是一个经常值得考虑的问题,如常说的“负负得正”。对于表达式“ $--5$ ”(为了与 C 语言的自减运算符区分,两个负号之间特地用了—个空格,这个表达式与 $-(-5)$ 等价,即等于 5),负号作为一个运算符,显然后面不能直接跟一个运算符,而应该跟一个运算量,这样一来,就应该先处理后面(右边)的负号,也就是说后面的负号与运算量 5 结合,得到一个 -5 ,在此基础上,再运算前一个(左边)的负号,得结果为 5。这里就存在一个先计算哪一个负运算的问题,显然两个运算符的优先级相同(同一个运算符),此时,应根据运算符的结合性来决定先运算哪一个,从上面的分析可以看出,负运算的结合性是自右向左。

在运算符的结合性不是很清楚的情况下,可以适当地添加小括号或添加空格来增强表达式的可读性。

2.2.3 复合赋值运算符

复合赋值运算符在一定程度上可以提高程序的编译效率,但会影响程序的可读性,尤其对于初学者,应该限制性地使用这类运算符。

在使用复合赋值运算符时,一定要弄清楚它们的含义,其作用是将左边的变量与右边的表达式进行运算,再将结果赋值给左边的变量,因此也称为自反运算。

当复合赋值运算符右边是一个表达式,而非单个变量时,需特别注意。如表达式 $a*=b/c$,该表达式相当于 $a=a*(b/c)$,而不能看作 $a=a*b/c$ 。

2.2.4 赋值类型转换

如果赋值运算符连接的左值与右值的数据类型不一致,在赋值前系统将自动把右值按左值的数据类型进行转换(也可以用强制类型转换的方式,把右值类型转换为左值类型后再进行赋值),但这种方式仅限于某些数据类型之间,通常称为“赋值兼容”。

常用的赋值转换规则如下。

- (1) 当右值为实型,左值为整型时,把实型数据的小数部分截断。
- (2) 当右值为整型,左值为实型时,数值不变,但以浮点数形式存储于左值中。
- (3) 当右值为 `double` 类型,左值为 `float` 类型时,取右值的前 7 位有效数字,赋值于左值。
- (4) 当右值为字符型,左值为整型时,将右值进行扩展(带符号数进行符号位扩展,无符号数进行 0 扩展),再赋值于左值。此规则同样适用于右值为较短的整型,而左值为较长的整型的情况。

(5) 当右值为整型, 左值为字符型时, 截取右值的低 8 位, 原封不动地赋值于左值。

2.3 例题分析

例 2.1 以下各项中不合法的用户标识符是()。

A. st.n B. file C. Main D. GO

解: C 语言规定标识符只能由字母、数字与下画线构成, 由此得出答案为选项 A, 因为它含有“.”符号。选项 B 是合法的标识符。选项 C、D 中包含了大写字母, 但 C 是区分字母大小写的, 所以 C、D 不是 C 语言的保留字符, 可以作为合法的标识符。

例 2.2 以下各项中不正确的实型常量是()。

A. 7.375E - 1 B. 0.3048e2 C. - 44.44 D. 123e - 2.5

解: 实型常量有小数形式与指数形式两种表示法。选项 C 是小数形式表示的实型常量, 是正确的。指数形式的实型常量由两部分组成, 即尾数与指数, 其中尾数必须是一个合法的小数形式的实型常量, 而指数必须是一个合法的整型常量。根据以上分析, 选项 D 的指数部分不是一个合法的整型常量, 即是不正确的实型常量, 故选 D。

例 2.3 下列不合法的十六进制整型常量是()。

A. 0xff B. 0Xabd C. 0x16 D. 0x345

解: C 语言规定十六进制整型常量以 0x 或 0X 开头, 数字符号范围为 0 至 9、'A'至'F'或'a'至'f'。选项 A 以字母 o 开头, 不符合 C 语言的规定, 是不合法的十六进制数, 故选 A。顺便指出, 以字母(o)开头恰好符合标识符的命名规则, 故 A 可以作为合法标识符, 但不是一个合法的十六进制常量。

例 2.4 以下程序的运行结果是()。

```
main()
{ int a=3, b=3, c, d;
  c=a++; d=b--; printf("%d,%d,%d,%d\n", a, b, c, d);
  c=++a; d=--b; printf("%d,%d,%d,%d\n", a, b, c, d);
}
```

A. 3, 3, 4, 2 B. 4, 3, 3, 3 C. 4, 2, 3, 2 D. 4, 2, 3, 3
5, 1, 5, 3 3, 3, 5, 3 5, 2, 5, 3 5, 1, 5, 1

解: 自增(减)运算符有前缀与后缀两种使用形式, 其区别是, 使用前缀方式时, 先实现变量的自增(减)1, 而使用后缀方式时, 先引用变量的当前值, 然后变量再自增(减)1。本题正确答案为 D。

例 2.5 若变量已正确定义并赋值, 以下符合 C 语言语法表达式的是()。

A. a+=b+c=5+d B. a=7+b+c, a++ C. a=a-8; D. 4.5%2

解: 对于选项 A 的后半部分“b+c=5+d”, 由于赋值运算符的左边出现了表达式, 故是错误的; 选项 C 看起来是正确的, 但因为后面有一个“;”, 所以不是表达式, 而是赋值语句, 故是错误的; 选项 D 中的%运算符要求参与运算的量为整型量, 故也是错误的; 只有选项 B 是正确的, B 中赋值符号的右侧是一个合法的逗号表达式。

例 2.6 写出下面程序的运行结果。

```
main()
{ int a=7,b=2;
  printf("%d,%d,%f,%f\n", a/b,a%b,(float)a/b, (float)(a/b));
}
```

解：程序的运行结果是：3,1,3.500000,3.000000。

因为 a 和 b 都是整型，所以 a/b 是整型，值为 3，不是 3.5；a%b 是求 a 除以 b 的余数，所以是 1；(float)a/b 是先将 a 转换成实型，再与 b 相除，b 当然也要转换成实型，所以结果是实型；(float)(a/b) 是先做 a/b 运算(值为 3)，然后再将 3 转换成实型。

例 2.7 写出下面程序的运行结果。

```
main()
{ printf("\%%\%101\102\103\104%\%\\\"n");
  printf("\x61\x62\x63\x64\\\"%\%n");
  printf("\ABCD\60\61\62\63\64\b\x30\x31\x32\x33\n");
}
```

解：程序的运行结果如下。

```
"\%ABCD%\%"
'abcd\'%
"ABCD"01230123
```

对于放在 printf 函数的双引号内的转义字符，(\")输出一个双引号，(\')输出一个单引号，(\\)输出一个反斜线，(\101)输出 A，(\102)输出 B，(\103)输出 C，(\104)输出 D，(\x61)至(\x64) 输出 abcd，(\60)至(\64) 输出 01234，(\b)是退格，退格的结果擦掉了 4，(\x30)至(\x33)输出 0123。对于放在 printf 函数的双引号内的其他字符(如 ABCD)原样输出。对于放在 printf 函数的双引号内的两个并列的%只输出一个%。注意(%%)与(%d)或(%c)等输入、输出控制符号不同。

2.4 习 题

2.4.1 单项选择题

- 数据在内存中是以()形式存放的。
A. 二进制 B. 八进制 C. 十进制 D. 十六进制
- 下列数中，()是浮点数的正确表示形式。
A. 223 B. 719E22 C. e23 D. 12e2.0
- 字符型常量在内存中存放的是它的()二进制形式。
A. ASCII 码 B. BCD 码 C. 内部码
- “BB\n\\123\t”包括的字符数是()。

- A. 6 B. 7 C. 8 D. 9
5. 若变量 a、i 已正确定义, 且 i 已正确赋值, 以下合法的语句是()。
- A. a==1 B. ++i; C. a=a++=5; D. a=int(i);
6. 下列运算符中, 结合方向为自左向右的是()。
- A. = B. , C. += D. --
7. 若有 “int x;”, 则 sizeof(x) 和 sizeof(int) 两种描述()。
- A. 都正确 B. 值不一样 C. 前者正确 D. 后者正确
8. 整型变量 x=1, y=3, 经下列计算后, x 的值不等于 6 的是()。
- A. x=(x=1+2, x*2) B. x=y%5/2*6
- C. x=9-(-y)-(++x) D. x=y*4.2/2
9. 单精度变量 x=3.0, y=4.0, 下列表达式中 y 的值为 9.0 的是()。
- A. y/=x*27/4 B. y+=x+2.0 C. y-=x+8.0 D. y*=x-3.0
10. 若整型变量 i=3, j=4, 整型变量 k=(i++)+(j--) 后, k 的值为()。
- A. 6 B. 7 C. 8 D. 9
11. 设有整型变量 x=1, 表达式 (x||0)&&(!x) 的值为()。
- A. 0 B. 1 C. 10 D. 11
12. 执行 “printf(“number:\101,\x42”);”, 输出为()。
- A. number:\101,\x42 B. number:A,B
- C. number:101,x42 D. number:a,b
13. 有整型变量 x, 单精度变量 y=5.5, 表达式 x=(float)(y*3+((int)y)%4) 执行后, x 的值为()。
- A. 17 B. 17.500000 C. 17.5 D. 16
14. 表达式 (a=10%7, b=3/2, a=a+b, a%3) 的值为()。
- A. 1.5 B. 1 C. 4 D. 3
15. 以下各项中值最大的是()。
- A. sizeof(int) B. sizeof(unsigned long)
- C. sizeof(double) D. sizeof(long)
16. 在 C 语言中, 不正确的 short int 类型的常数是()。
- A. 32768 B. 0 C. 037 D. 0xAF
17. 当变量 c 的值为 0 时, 在下列选项中能正确将 c 的值赋给变量 a、b 的是()。
- A. c=b=a; B. (a=c)||(b=c);
- C. (a=c)&&(b=c); D. a=c=b;
18. 下列程序的输出结果是()。
- ```
main()
{ double d=3.2; int x, y; x=1.2; y=(x+3.8)/5.0;
 printf(“%d\n”, d*y);
}
```
- A. 0                      B. 3.2                      C. 3                      D. 3.07

19. 下列各式中,合法的变量定义形式是( )。
- A. short \_a=.1e-1;      B. double b=1+5e2.5;  
C. long do=0xfdaL;      D. float 2\_and=1-e-3;
20. 设“int x=1,y=1;”,表达式(--x||y--)值是( )。
- A. 0      B. 1      C. 2      D. -1
21. 下列程序执行后的输出结果是( )。

```
main()
{ int x='f'; printf("%c\n", 'A'+(x-'a'+1)); }
```

- A. G      B. H      C. I      D. J

22. 下列程序执行后的输出结果是( )。

```
main()
{ int x=-1; printf("%d, %o, %x\n", x, x, x); }
```

- A. -1, -1, -1      B. -1, 65535, ffff  
C. -1, 177777, ffff      D. -32768, -1, 65535

23. 设有如下定义。

```
int i=8, k, a, b;
unsigned long w=5;
double x=1.42, y=5.2;
```

则以下符合 C 语言语法的是( )表达式。

- A. a += a - =(b = 4)\*(a = 3)      B. x%(-3)  
C. a = a \* 3 = 2      D. y = float(i)

24. 以下程序段的输出结果是( )。

```
int a=2,b=5;
printf("a=%d,b=%d\n", a,b);
printf("a=%d,b=%d\n", a,b);
```

- A. a=%2,b=%5      B. a=2,b=5      C. a=%d,b=%d      D. a=%d,b=%d  
a=2,b=5      a=2,b=5      a=2,b=5      a=2,b=5

25. 执行“printf("\%d\141\40\142%c\"x31\40\x32\x33",3+4,'e');”,输出为( )。

- A. \\7A Be"1 23      B. \\7a be"1 23  
C. \\7A Be"1 23      D. \\7a be"1 23

26. 以下程序的输出结果是( )。

```
main()
{ int a=3; printf("%d\n",a+(a-=a*a)); }
```

- A. -6      B. 12      C. 0      D. -12

27. 运行以下程序时,从键盘输入 567,输出是( )。

```
int main()
```

```

{ int a,b,c,d,e; scanf("%d",&a);
 b=a/100; c=(a-b*100)/10; d=(a%100)%10; e=d*100+c*10+b;
 printf("%d\n",e);
}

```

A. 567            B. 756            C. 576            D. 765

28. 若以下所有变量已正确定义并赋值, 下面符合 C 语言语法表达式的是( )。

A. a:=b+1        B. a=b=c+2        C. int a=18.5%3        D. a=a+7=c+b

29. 若已定义 x 和 y 为 double 类型, 则表达式(x=1, y=x+3/2)的值是( )。

A. 1            B. 2            C. 2.0            D. 2.5

30. 运行如下程序的输出结果是( )。

```

main()
{ int y=3,x=3,z=1; printf("%d %d\n",(++x,y++),z+2); }

```

A. 3 4            B. 4 2            C. 4 3            D. 3 3

## 2.4.2 填空题

1. C 语言规定, 标识符只能由\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_这 3 种字符组成, 而且, 第一个字符必须是\_\_\_\_\_或\_\_\_\_\_。

2. 一个 C 程序一般由若干函数组成, 程序中有且只有一个\_\_\_\_\_。

3. 一个 C 函数是由\_\_\_\_\_和\_\_\_\_\_这两部分组成的。

4. 一个 C 程序总是从\_\_\_\_\_函数开始执行的。

5. C 语言的基本数据类型有\_\_\_\_\_、\_\_\_\_\_和实型, 其中实型又分为\_\_\_\_\_和\_\_\_\_\_。

6. 在内存中占据 16 位的无符号整型变量的取值范围是\_\_\_\_\_到\_\_\_\_\_。

7. 在 C 语言中, 八进制整型常量以\_\_\_\_\_开头, 十六进制整型常量以\_\_\_\_\_开头。

8. 十进制 short int 型数-1 转换为十六进制 short int 型数为\_\_\_\_\_。

9. 在 C 程序中, %运算符可用作\_\_\_\_\_运算, 而\_\_\_\_\_运算可用++作为运算符, sizeof(float)是求\_\_\_\_\_。

10. 将运算符“/、+、++、,、%”按优先级从高到低的顺序排列是\_\_\_\_\_。

11. 'x'在内存中占\_\_\_字节, "x"在内存中占\_\_\_字节, "\101"在内存中占\_\_\_字节。

12. 若 x 为整型变量, 执行语句“x='b'-'A';”后, x 的值为\_\_\_\_\_。

13. 单精度实型变量 x, 执行语句“x=8.4+13.1\*2-'a'%7;”后, x 的值为\_\_\_\_\_。

14. 整型变量 a、b、c、i、j, 执行表达式语句“i=3; a=i++; b=i++; c=i++; j=a+b+c;”后, 变量 i 的值为\_\_\_\_\_, 变量 j 的值为\_\_\_\_\_。

15. 整型变量 a、b、c、d 的初值都是 3, 计算表达式 d=(++a,a++,b=a--,c=(++b)+(-c))后, a、b、c 的值分别为\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, d 的值为\_\_\_\_\_。

16. 已知有 a、b 两个数, 执行“x=b; b=a; a=x;”操作后, 执行的效果是\_\_\_\_\_。

17. 将下列数学式改写成 C 语言的表达式。

(1)  $ax^3+bx^2+cx+d$  可以写成\_\_\_\_\_。

(2)  $\sqrt{\frac{a^2+b^2}{a*b}}$  可以写成\_\_\_\_\_。

18. 执行如下程序:

```
main()
{ int x=0, y=0, z=0;
 x=(y=y+1)+(z=z+1); printf("%d,%d,%d*",x,y,z);
 x=(++y, ++z); printf("%d,%d,%d*",x,y,z);
 x+=(y=y-1)*(z=1); printf("%d,%d,%d*",x,y,z);
}
```

程序输出结果为\_\_\_\_\_。

19. 整型变量 a=5, b=7, 则表达式 b/a \* 100 的值是\_\_\_\_\_。

20. 以下程序段运行的结果是\_\_\_\_\_。

```
int a=5, b=3;
printf("a+b=%d ", a+=b+=a+b); /*执行本语句后, b 变为 11, a 变为 16 */
printf("a-b=%d\n", a=b-a-b);
```

### 2.4.3 阅读程序写结果题

1. #include "stdio.h"

main()

```
{ int a,b,d=241; a=d/100%9; b=(-3)%(-2); printf("%d,%d",a,b); }
```

2. #include "stdio.h"

main()

```
{ int i,j,x,y; i=5;j=7; x=++i; y=j++;
 printf("%d,%d,%d,%d",i,j,x,y); }
```

3. #include "stdio.h"

main()

```
{ float f=13.8; int n; n=((int) f)%3; printf("n=%d",n); }
```

4. #include "stdio.h"

main()

```
{ int n=2; n+=n-=n*n; printf("n=%d",n); }
```

5. #include "stdio.h"

main()

```
{ int a,b,x; x=(a=3,b=a--); printf("x=%d, a=%d, b=%d",x,a,b); }
```

6. #include "stdio.h"

main()

```
{ float f1,f2,f3,f4; int m1,m2; f1=f2=f3=f4=2; m1=m2=1;
 printf("%d,%d\n",m1+=f1*f2/(f3*f4),m2+=f1*f2/f3*f4);
}
```

```
7. #include "stdio.h"
main()
{ int x=5,y=2; printf("1:%d; ",y=x/2); printf("2:%d;", y=x%2);
 printf("3:%d;", y=(float)(x/2)); printf("4:%d", y=(float)x/2);
}
```

```
8. #include "stdio.h"
main()
{ unsigned short b=65535; printf("%d, %u",b, b); }
```

```
9. #include "stdio.h"
main()
{ int a,b,c,x=10,y=9;
 a=(-x / y++); b=++y; c=b+(-x)%y;
 printf("a=%d, b=%d, c=%d\n", a, b, c);
}
```

```
10. #include "stdio.h"
main()
{ int x=0,y=0,z=0;
 x=++y+(++z); printf("%d,%d,%d*",x,y,z);
 x=(y++)+z++; printf("%d,%d,%d*",x,y,z);
 x=(-y)+(z--); printf("%d,%d,%d*",x,y,z);
}
```

```
11. #include "stdio.h"
main()
{ short i=-1; printf("%d,%o,%x,%u\n",i,i,i,i); }
```

```
12. #include "stdio.h"
main()
{ short i=1; printf("%d,%o,%x,%u\n",i,i,i,i); }
```

```
13. #include "stdio.h"
main()
{ char c='A'; printf("%d,%o,%x,%c\n",c,c,c,c); }
```

```
14. #include "stdio.h"
main()
{ float f=3.1415927; printf("%f,%5.4f,%3.3fn",f,f,f); }
```

```
15. #include "stdio.h"
main()
{ float f=3.5; printf("%f,%g\n",f,f); }
```

### 2.4.4 编写程序题

1. 利用变量  $k$ ，将两个变量  $m$  和  $n$  的值进行交换。
2. 输入一个三位数  $n$ ，将其各位数码逆序输出(如输入 672，输出 276)。
3. 输入一个整数  $n$ ，输出  $n$  除以 3 的余数。
4. 输入一个英文字符，分别输出它的十进制、八进制、十六进制的 ASCII 码值。
5. 输入两个实数，计算这两个实数的平均值，分别输出保留 2 位或 4 位小数的平均值。
6. 输入一个三位整数  $n$ ，求  $n$  的三位数码之和。
7. 编写一个程序，要求不使用中间变量，实现两个变量  $a$ 、 $b$  值的交换。

## 2.5 习题参考答案

### 2.5.1 单项选择题答案

- |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1. A  | 2. B  | 3. A  | 4. A  | 5. B  | 6. B  | 7. A  | 8. C  | 9. B  | 10. B |
| 11. A | 12. B | 13. A | 14. B | 15. C | 16. A | 17. B | 18. C | 19. A | 20. B |
| 21. A | 22. C | 23. A | 24. D | 25. D | 26. D | 27. D | 28. B | 29. C | 30. D |

### 2.5.2 填空题答案

- |                      |                     |             |    |              |                                    |
|----------------------|---------------------|-------------|----|--------------|------------------------------------|
| 1. 字母                | 数字                  | 下画线         | 字母 | 下画线          | 2. main 函数                         |
| 3. 函数首部              |                     | 函数体         |    |              | 4. main                            |
| 5. 整型                | 字符型                 | 单精度实型 float |    | 双精度实型 double |                                    |
| 6. 0                 | $2^{16}-1$          | 7. 0        | 0x |              | 8. 0xFFFF                          |
| 9. 求余数               | 自增                  | 单精度实型所占的字节数 |    |              | 10. ++、/、%、+、,                     |
| 11. 1                | 2                   | 2           |    | 13. 28.6     | 14. 6 12                           |
| 15. 4                | 6                   | 8           | 8  |              | 16. 交换 a,b 的值                      |
| 17. (1)              | $a*x*x+b*x*x+c*x+d$ |             |    |              | (2) $\text{sqrt}((a*a+b*b)/(a*b))$ |
| 18. 2,1,1*2,2*3,1,1* | 19. 100             |             |    |              | 20. $a+b=16$ , $a-b=4$             |

### 2.5.3 阅读程序写结果题答案

- |                      |                           |                      |
|----------------------|---------------------------|----------------------|
| 1. 2, -1             | 2. 6,8,6,7                | 3. $n=1$             |
| 4. $n=-4$            | 5. $x=3, a=2, b=3$        | 6. 2,5               |
| 7. 1:2;2:1;3:2;4:2   | 8. -1,65535               | 9. $a=1, b=11, c=19$ |
| 10. 2,1,1*2,2*3,1,1* | 11. -1,177777,ffff,65535  | 12. 1,1,1,1          |
| 13. 65,101,41,A      | 14. 3.141593,3.1416,3.142 | 15. 3.500000,3.5     |

### 2.5.4 编写程序题参考答案

1. 

```
main()
{ int m,n,k; scanf("%d,%d",&m,&n); /*如果输入 3, 5*/
```

```
k=m; m=n; n=k; printf("%d,%d",m,n); /*那么输出 5, 3*/
}
```

2. main()

```
{ int n, a,b,c;
printf("输入一个三位数: "); scanf("%d",&n);
a=n/100; b=n/10%10; c=n%10; printf("%d%d%d",c,b,a);
}
```

3. main()

```
{ int n; scanf("%d",&n); printf("%d",n%3); }
```

4. main()

```
{ char a; scanf("%c",&a); printf("%d,%o,%x\n",a,a,a); }
```

5. main()

```
{ float a,b,c; scanf("%f,%f",&a,&b);
c=(a+b)/2; printf("%9.2f,%9.4f\n",c,c);
}
```

6. main()

```
{int n, p,s=0; scanf("%d",&n); /*若 n 为 123*/
p=n%10; s=s+p; n=n/10; /*n 变为 12*/
p=n%10; s=s+p; n=n/10; /*n 变为 1*/
p=n%10; s=s+p;
printf("%d",s);
}
```

7. main()

```
{int a=3,b=5; printf("%d , %d",a,b); /*输出 3, 5*/
a=a+b; b=a-b; a=a-b;
printf("%d , %d", a,b); /*输出 5, 3*/
}
```