

21 世纪高等院校计算机系列课程教材

C 语言程序设计 习题与上机指导

主编 高 巍 王淮中

(第2版)



北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

21 世纪高等院校计算机系列课程教材

C 语言程序设计 习题与上机指导

(第 2 版)

主编 高 巍 王淮中
编者 姜 楠 张丽秋 张 颜
张立忠 梅 旃



 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

现阶段很多高等院校把《C 语言程序设计》课程设置为必修课,学生在学习了本课程之后,大部分还要参加全国计算机等级考试。为了能让广大初学者在短期内迅速掌握 C 语言,同时为课程相关实验提供配套指导书,特编写本书。

本书共分为两部分,第一部分为教材各个章节的专项练习题,可以帮助读者复习和巩固所学知识;第二部分为实验指导,循序渐进地完成 18 个实验,逐步掌握在 VC++6.0 下调试程序技巧,初步掌握程序设计的算法。

本书内容丰富,实用性强,不仅可以作为自学者的参考资料,而且可以作为其他各类 C 语言教材的辅助教材,也可供各类报考 C 语言考试的读者参考。

版权专有 侵权必究

图书在版编目(CIP)数据

C 语言程序设计习题与上机指导/高巍,王淮中主编. —2 版. —北京:北京理工大学出版社,2010.1

ISBN 978-7-5640-0316-6

I. C… II. ①高…②王… 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(2009)第 086235 号

出版发行/北京理工大学出版社

社 址/北京市海淀区中关村南大街 5 号

邮 编/100081

电 话/(010) 68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址/<http://www.bitpress.com.cn>

经 销/全国各地新华书店

印 刷/保定市中国画美凯印刷有限公司

开 本/787 毫米×1092 毫米 1/16

印 张/16.5

字 数/365 千字

版 次/2010 年 1 月第 2 版 2010 年 1 月第 5 次印刷

印 数/20001~24000 册

定 价/33.00 元

责任校对/陈玉梅

责任印制/边心超

图书出现印装质量问题,本社负责调换

前 言

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言。近年来，C 语言在国内得到迅速的推广应用，许多高校相继开设了《C 语言程序设计》课程。为了提高本课程的教学质量，改善客观上 C 语言难讲、难学、难以掌握的现状，因此编写了本书。

本书是《C 语言程序设计》教材的配套参考书，共分为两部分：第一部分为教材各章节专项的练习题，在每部分练习题之前，都给出了所涉及的知识点，并给出典型例题和详细的讲解；第二部分为上机实验指导，详细指明了学习中应该达到的具体目标，并结合具体实例对程序调试手段做了详细阐述，旨在为课堂教学过渡到独立编程提供更佳的模拟环境和知识拓展。本书中的大部分编程都给出了运行结果，用以印证编程的思路，体现程序的意图，而且由结果反观程序也能更好地理解程序的功能，更有助于从细微处体会编程的技巧。如果能对这些习题细加揣摩，那么读者的编程能力定会有所提高。

程序设计是创造性劳动，它需要编程人员全身心投入和主观能动性的充分发挥。书中的程序都是编者精心选取和编写的，全部在 VC++6.0 下调试通过。本书虽然是教材的配套参考书，但完全可以单独使用，有些程序具有很强的实用性，读者可以从中得到启发。

我们的愿望是好的，工作是努力的，但限于水平、能力和时间，书中存在不妥之处在所难免，恳请读者同仁批评指正。

编者

目 录

第一部分 练习题

第1章 C语言概述	1
1.1 知识点	1
1.2 典型题分析	1
1.3 习题	2
第2章 数据类型	4
2.1 知识点	4
2.2 典型题分析	4
2.3 习题	6
第3章 运算符与表达式	10
3.1 知识点	10
3.2 典型题分析	11
3.3 习题	12
第4章 语句	24
4.1 知识点	24
4.2 典型题分析	24
4.3 习题	28
第5章 数组	56
5.1 知识点	56
5.2 典型题分析	56
5.3 习题	60
第6章 函数	81
6.1 知识点	81
6.2 典型题分析	82
6.3 习题	86
第7章 指针	102
7.1 知识点	102
7.2 典型题分析	102
7.3 习题	107
第8章 预编译	139
8.1 知识点	139
8.2 典型题分析	139

8.3 习题	142
第 9 章 结构体与共用体	147
9.1 知识点	147
9.2 典型题分析	148
9.3 习题	150
第 10 章 位运算	155
10.1 知识点	155
10.2 典型题分析	155
10.3 习题	156
第 11 章 文件	160
11.1 知识点	160
11.2 典型题分析	161
11.3 习题	164
练习题参考答案	170

第二部分 实验指导

概述	213
实验 1 熟悉 C 语言程序的集成环境	213
实验 2 语法符号和数据类型	217
实验 3 标准输入函数和输出函数	219
实验 4 运算符和表达式	221
实验 5 顺序结构程序设计	223
实验 6 选择结构程序设计	225
实验 7 循环结构程序设计 (一)	228
实验 8 循环结构程序设计 (二)	230
实验 9 数组 (一)	232
实验 10 数组 (二)	235
实验 11 函数 (一)	238
实验 12 函数 (二)	241
实验 13 指针 (一)	243
实验 14 指针 (二)	246
实验 15 编译预处理	248
实验 16 结构体与共用体	250
实验 17 位运算	252
实验 18 文件	254

第一部分 练习题

第1章 C语言概述

1.1 知识点

1. C语言的发展历史

2. C语言程序范例及其结构特点

C语言程序的基本结构如下：

- 所有的C语言程序都由一个或多个函数组成。
- 在所有函数中，至少包含一个名为 `main()` 的主函数。
- C语言程序总是从主函数 `main()` 开始执行，`main()` 函数可以放在程序的任何位置。

3. C语言程序的开发过程

C语言程序的开发步骤：

- (1) 编辑源程序。使用一个文本编辑器编辑C语言源程序。
- (2) 编译。使用一个C语言编译系统对C语言源程序进行语法检查和翻译，生成同名的“.obj”目标文件。
- (3) 连接。将目标文件和库函数等连接在一起生成一个同名的“.exe”可执行文件。
- (4) 运行。运行生成的可执行文件。

也可以使用集成开发环境编程，例如 Turboc 2.0 或者 Microsoft Visual C++ 6.0。读者可参考本书实验部分的实验一。

1.2 典型题分析

【例题1】以下说法中正确的是（ ）。

- ☒ A. C语言程序总是从第一个函数开始执行
- ☐ B. 在C语言程序中，要调用的函数必须在 `main()` 函数中定义
- ☒ C. C语言程序总是从 `main()` 函数开始执行
- ☐ D. C语言程序中的 `main()` 函数必须放在程序的开始部分

【解析】

一个 C 语言程序是由一个或多个函数构成的,有且只有一个 main 函数,它是整个程序的入口,即程序执行的起点。因此选 C。

【例题 2】下列程序中错误的语句是 ()。

- A. void main() B. { int a, b, z; C. a=2 : b=3; D. z=a+b; }

【解析】

选 C。C 语言程序由多个语句构成,每条语句代表一个独立的运行步骤,每条语句以“;”号结束,分号是语句不可缺少的组成部分。因此应将 C 项改为 a=2; b=3;。

1.3 习题

一、选择题

- 构成 C 语言程序的基本单位是 ()。
A. 过程 B. 子程序 C. 函数 D. 子函数
- 下列关于 main 函数的叙述正确的是 ()。
A. C 语言中的 main 函数必须位于程序的最前面
B. main 函数的功能主要是调用其他函数
C. C 语言程序中的 main 函数与其他函数的地位不一样
D. C 语言程序中可以没有 main 函数
- C 语言中,复合语句要用哪一种括号括起来 ()。
A. 方括号 B. 圆括号 C. 花括号 D. 尖括号
- 下列叙述不正确的是 ()。
A. 一个 C 语言程序可由一个或多个函数组成
B. 一个 C 语言程序必须包含一个 main 函数
C. C 语言程序的基本组成单位是函数
D. 在 C 语言程序中,注释说明只能位于一条语句的后面
- C 语言规定:在一个源程序中,main 函数的位置 ()。
A. 必须在最开始 B. 必须在系统调用的库函数的后面
C. 可以任意 D. 必须在最后
- 下列叙述正确的是 ()。
A. C 语言程序由主函数组成 B. C 语言程序由主函数和过程组成
C. C 语言程序由函数组成 D. C 语言程序由函数和过程组成
- 一个 C 语言程序的执行是从 ()。
A. 本程序的 main 函数开始,到 main 函数结束
B. 本程序文件的第一个函数开始,到本程序文件的最后一个函数结束
C. 本程序的 main 函数开始,到本程序文件的最后一个函数结束
D. 本程序文件的第一个函数开始,到本程序 main 函数结束
- 一个 C 语言程序中 ()。
A. 必须有一个主函数 B. 可以有多个主函数

- C. 必须有主函数和其他函数 D. 可以没有主函数
9. 下列叙述正确的是 ()。
- C. A. C 语言程序中, 注释部分两侧的分界符分别为 `/*` 和 `*/`
B. C 语言程序的每行中只能写一条语句
C. C 语言本身没有输入/输出语句
D. 在对一个 C 语言程序进行编译的过程中, 可发现注释中的拼写错误
10. 下面程序中不正确的是 ()。
- ```
void main()
{
 A. {int a, b, sum; B. a=12;
 C. b=23; D. c=a+b; }
```
11. C 语言具有低级语言的能力, 主要指的是 ( )。
- A. 程序的可移植性  
B. 具有控制流语句  
C. 能直接访问物理地址, 可进行位操作  
D. 具有现代化语言的各种数据结构
12. 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

## 二、编程题

1. 试编写一个 C 语言程序, 输出以下信息:

\*\*\*\*\*

Very good!

\*\*\*\*\*

2. 试编写一个程序, 能显示出以下两行文字:

I am a student.

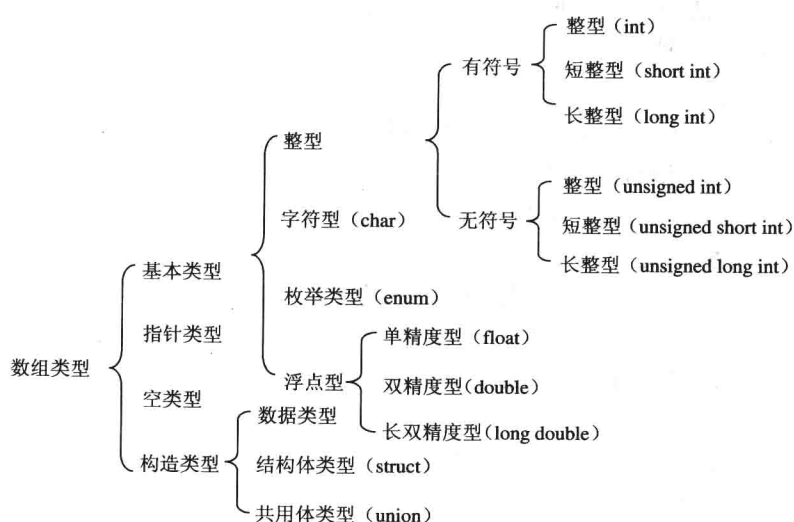
I love China.

3. 试编写一个程序, 输入 a, b, c 三个值, 输出其中最大值。

## 第2章 数据类型

### 2.1 知识点

#### 1. 数据类型的分类



#### 2. 常量和变量

C 语言中的数据分为常量和变量，它们分别属于以上这些类型。在程序运行过程中，其值不能改变的量称为常量，其值可以改变的量称为变量。

#### 3. 标准输入输出函数

printf()函数与 scanf()函数。

### 2.2 典型题分析

【例题 1】下列属于 C 语言提供的合法关键字是 ( )。

A. Int

B. float

C. VOID

D. include

【解析】

选 B。关键字是由系统预定义的词法符号，C 语言一共有 32 个关键字，所有关键字均为小写。

【例题 2】下列标识符中，合法的用户标识符是 ( )。

A. sum

B. char

C. 7\_x

D. b+a

**【解析】**

选 A。标识符是用户按命名规则自行定义的词法符号。标识符命名规则：标识符由一个或多个字母、数字或下划线组成，因此 D 错；标识符的第一个字符必须是字母或下划线，因此 C 错；标识符不能和关键字相同，因此 B 错。

**【例题 3】** 下列常量中不合法的常量是 ( )。

A. '\2'

B. "#"

C. 'abc'

D. "483"

**【解析】**

选 C。注意字符常量和字符串常量的区别。字符常量是用单引号括起来的单个字符，因此 C 错；字符串常量是一对双引号括起来的字符序列。A 是一个正确的转义字符，B 和 D 是正确的字符串常量。

**【例题 4】** 下列合法的转义字符是 ( )。

A. '\"'

B. '\018'

C. '\ee'

D. 'xab'

**【解析】**

选 A。转义字符以“\”开头，可在“\”后跟随 1~3 位八进制数，一个八进制数应由 0~7 共 8 个基本数码组成（因此 B 错）；也可在“\x”后跟随 1~2 位十六进制数（因此 C、D 错）。A 项代表双引号字符(“)。

**【例题 5】** 下列变量定义语句中，正确的是 ( )。

A. float a,b,c;

B. int a=b=c=5;

C. int x;y;

D. int x,y,

**【解析】**

选 A。C 语言规定“变量要先定义后使用”。在程序中用到的每一个变量都要先定义它属于哪一种类型。A 正确，a, b, c 三个变量被定义成实型变量，变量之间用逗号分隔，因此 C 错。变量在定义的同时可以赋值，但 B 项赋值语法形式不对，应改为 int a=5,b=5,c=5;和其他语句一样，变量定义语句后应以分号结束，因此 D 错。

**【例题 6】** 若 x,y 均定义为 int 型，z 定义为 double 型，下列不合法的 scanf 语句是 ( )。

A. scanf("%d,%lx,%lf",&amp;x,&amp;y,&amp;z);

B. scanf("%2d\n%d%lf",&amp;x,&amp;y,&amp;z);

C. scanf("\t%x,%o",&amp;x,&amp;y);

D. scanf("%x,%o,%6.2f",&amp;x,&amp;y,&amp;z);

**【解析】**

选 D。scanf 函数中的格式串用来描述输入项的数据类型和输入格式。和 printf 函数一样，scanf 函数可在类型描述字符 d,o,x,u 前加上前缀 l 或 L，表示把待输入项按长整型数据输入。在 f 前面加上前缀 l 或 L 表示输入 double 类型的数据，因此 A 正确。scanf 函数中允许出现空白符，如空格、制表符“\t”、换行符“\n”因此 B 和 C 正确。C 语言允许指定输出数据的小数点位数，系统将自动截取所需数据，如 printf("%3d,%6.2f",&x,&y)，但是 scanf 函数中不允许出现 %m.nf 格式，因此 D 错误。

**【例题 7】** 执行下面几个语句后，输出的结果为 ( )。

```
int i;
```

```
char c;
```

```
i='a';
```

```
c=97;
```

```
printf("%c,%d",i,c);
```

- A. 97,a      B. a,97      C. a,a      D. 97,97

### 【解析】

选 B。在 C 语言中,字符型数据和整型数据可以相互赋值。字符型数据或整型数据既可以用字符形式(%c)输出,也可以用整数形式(%d)输出。本题中,如果用格式符%c将整型变量 i 输出可得到字符 a,用%d输出 i 可得到'a'对应的 ASCII 码值 97。同理,用%d将字符型变量 c 输出可得到 97,用%c输出变量 c 可得到 ASCII 码值为 97 对应的字符 a。

【例题 8】对于下述语句,分别将 100 和 200 赋给变量 x 和 y,则应按( )方式输入数据。

```
int x,y;
```

```
scanf("%d,%d",&x,&y);
```

- A. 100,200      B. 100200      C. 100 200      D. 100;100

### 【解析】

选 A。使用 scanf 函数时,输入内容的格式必须与格式串中规定的格式相同。由于格式串“%d,%d”中含有一个逗号,则 scanf 函数在读入第 1 个整数存入 x 后需要读入一个逗号,只有得到逗号后才继续读入第 2 个整数并存入 y 中。若此时读入的不是逗号而是其他字符,如空格、分号、冒号等,则第 2 个整数不能被正确地读入到 y 中。

## 2.3 习题

### 一、选择题

- C 语言中, double 类型数据占( )个字节。  
A. 1      B. 2      C. 4      D. 8
- C 语言中, char 类型数据占( )个字节。  
A. 1      B. 2      C. 4      D. 8
- ( )是 C 语言中提供的合法的数据类型关键字。  
A. Float      B. signed      C. integer      D. Char
- 在下列标识符中,合法的用户标识符是( )。  
A. xy\_      B. s\*.1      C. 0\_t      D. k%
- 在下列标识符中,合法的用户标识符是( )。  
A. Fast+Big      B. void      C. Pb1      D. <book>
- 以下选项中不合法的用户标识符是( )。  
A. abC.x      B. file      C. Main      D. PRONTF
- 下列字符序列中,不可用作 C 语言标识符的是( )。  
A. b70      B. #Ab      C. \_SYMBOL      D. a\_1
- 下列正确的浮点常量是( )。  
A. E2      B. 1.5e2.3      C. 5.0e      D. 3e-3
- 下列错误的常量是( )。  
A. 5.      B. 0.123      C. 090      D. 0xf

10. 下列形式的常数中, C 程序不允许出现的是 ( )。

- A. .45      B.  $\pm 123$       C.  $25.6e-2$       D.  $4e3$

11. 下列合法的常量是 ( )。

- A. 1.52e      B.  $\pm 12$       C. "x"      D. "s" 'y'

12. 下列正确的字符常量是 ( )。

- A. "a"      B. "\\"      C. 'W'      D. ''

13. 下列数据中, 正确的字符串常量是 ( )。

- A. 'A'      B. "house"      C. How do you do      D. '\$abc'

14. 下面不正确的字符串常量是 ( )。

- A. 'abc'      B. "12'12"      C. "0"      D. " "

15. 在 C 语言中, 字符型数据在内存中以 ( ) 形式存放。

- A. 原码      B. BCD 码      C. 反码      D. ASCII 码

16. 下列变量定义语句中, 正确的是 ( )。

- A. char: a,b,c;      B. char x;y;z;      C. int t;k;      D. float w,m;

17. 执行下面两个语句后, 输出的结果为 ( )。

```
char c1=97,c2=98;
printf("%d,%c",c1,c2);
```

- A. 97,98      B. 97,b      C. a,98      D. a,b

18. 在 C 语言中, int 类型数据占两个字节, 则下列语句的输出结果为 ( )。

```
int k=-1;
printf("%d,%u\n",k,k);
```

- A. -1,-1      B. -1,32767      C. -1,32768      D. -1,65535

19. 以下程序的输出结果为 ( )。

```
main()
{ printf("%e",123.456);}
```

- A. 123.456      B. 123.4560      C. 1.23456e2      D. 1.23456e+02

20. 若 x 为 float 型变量, 则以下语句 ( )。

```
x=213.82631;
printf("%-4.2f\n",x);
```

- A. 输出格式描述符的域宽不够, 不能输出      B. 输出为 213.83  
C. 输出为 213.82      D. 输出为 -213.82

21. 设 a=12, b=12345; 执行语句 printf("%4d, %4d",a,b) 的输出结果为 ( )。

- A. 12, 123      B. 12, 12345      C. 12, 1234      D. 12, 123456

22. 执行下述程序段时输出的结果是 ( )。

```
float x=-1023.012;
printf("\n%8.3f",x);
printf("%10.3f",x);
```

- A. 1023.012    -1023.012      B. -1023.012,    -1023.012  
C. 1023.012,    -1023.012      D. -1023.012    -1023.012

23. 对于下述语句, 分别将 56 和 78 赋给变量 y1 和 y2, 则应按 ( ) 方式输入数据。

```
int y1,y2;
```

```
scanf("y1=%d,y2=%d",&y1,&y2);
```

- A. y1=56y2=78    B. 56,78    ☒ C. y1=56,y2=78    D. 56 78

24. 对于下述语句, 分别将 12 和 34 赋给变量 x1 和 x2, 则应按 ( ) 方式输入数据。

```
int x1,x2;
```

```
scanf("%d%d",&x1,&x2);
```

- ☒ A. 1234    ☒ B. 12 34    ☒ C. 12,34    D. 12:34

25. 假设定义变量如下:

```
int x;
```

```
float y;
```

则下列输入语句中正确的是 ( )。

- A. scanf("%f%f",&y,&x);    B. scanf("%f%d",y,x);  
☒ C. scanf("%f,%d",&y,&x);    D. scanf("%5.2f%2d",&y,&x);

26. 有如下定义:

```
float x;
```

```
unsigned y;
```

则合法的输入语句是 ( )。

- A. scanf("%5.2f%d",&x,&y);    ☒ B. scanf("%f%3o",&x,&y);  
☒ C. scanf("%f%n",&x,&y);    ☒ D. scanf("%f%f",&x,&y);

## 二、判断题

1. C 语言以函数为程序的基本单位, 便于实现程序的模块化。 ( ☒ )
2. C 语言程序中可以不使用函数。 ( ☒ )
3. C 语言程序的执行总是从程序的第一句开始。 ( ☒ )
4. 字符常量和字符串常量一样, 都是用双引号括起来的。 ( ☒ )
5. 字符常量存储时, 自动在字符后加一个字符结束标志 '\0'。 ( ☒ )

## 三、阅读程序并写出程序的运行结果

1. 以下程序的执行结果是: abc cc

```
#include "stdio.h"
```

```
void main()
```

```
{ char c1='a', c2='b', c3='c';
```

```
printf("a%cb%c\tc%c\n",c1,c2,c3);}
```

2. 以下程序的执行结果是: 24, 20, 36

```
#include "stdio.h"
```

```
void main()
```

```
{ int x,y,z;
```

```
x=24;
```

```
y=024;
```

```
z=0x24;
```

```
printf("%d,%d,%d",x,y,z);}
```

3. 以下程序的执行结果是：

A. B

```
#include "stdio.h"
```

```
void main()
```

```
{ char c1,c2;
```

```
 c1='a';
```

```
 c2='b';
```

```
 c1=c1-32;
```

```
 c2=c2-32;
```

```
printf("%c %c",c1,c2);}
```

# 第3章 运算符与表达式

## 3.1 知识点

C 语言的运算符丰富，表示内涵和范围广泛，掌握和灵活应用 C 语言的运算符，关键要注意以下几点：

- (1) 运算符的运算含义和运算规则；
- (2) 运算符的优先级和结合性；
- (3) 运算符对运算对象类型的要求和限制，这涉及运算对象类型的自动转换和强制类型转换的概念。表 3.1 列出了 C 语言运算符的优先级和结合性。

表 3.1 运算符的优先级与结合性

| 优先级 | 运 算 符                                        | 结合方向 |
|-----|----------------------------------------------|------|
| 1   | ( ) [ ] -> .(结构体成员运算符)                       | 从左向右 |
| 2   | ! ~ ++ -- - (类型转换) *(指针) &(取地址) sizeof       | 从右向左 |
| 3   | *(乘法) / %                                    | 从左向右 |
| 4   | +(加法) -(减法)                                  | 从左向右 |
| 5   | <<(左移) >>(右移)                                | 从左向右 |
| 6   | < <= > >= (关系运算符)                            | 从左向右 |
| 7   | == !=                                        | 从左向右 |
| 8   | &(按位与)                                       | 从左向右 |
| 9   | ^(按位异或)                                      | 从左向右 |
| 10  | (按位或)                                        | 从左向右 |
| 11  | &&(逻辑与)                                      | 从左向右 |
| 12  | (逻辑或)                                        | 从左向右 |
| 13  | ?: (条件运算符)                                   | 从右向左 |
| 14  | = += -= *= /= %= >>= <<= &= ^=<br> = (赋值运算符) | 从右向左 |
| 15  | ,(逗号运算符)                                     | 从左向右 |

掌握和应用 C 语言运算符的要点和难点需要注意 C 语言中的一些特殊运算符（如，++和--运算、关系运算符等）的运算规定。

运算符和操作符构成了表达式。一个表达式中可以包含多种不同的运算符，称之为复合表达式，对复合表达式的计算，需要充分考虑不同运算符的优先级和结合性，如果有两部分的优先级相同，则要根据结合性来运算。

## 3.2 典型题分析

【例题 1】下述程序的输出结果是（ ）。

```
#include "stdio.h"
void main()
{
 int x=1,y,z;
 y=++x;
 z=x++;
 printf("%d,%d,%d",x,y,z);
}
```

A. 3,2,2

B. 3,2,3

C. 2,2,2

D. 2,2,1

【解析】

选 A。 $++$ 运算符出现在变量  $x$  前，表示先将变量  $x$  值增加 1 后再引用它的值； $++$ 运算符出现在变量  $x$  后，表示先引用变量  $x$  值后，变量  $x$  再自增 1。本例中  $y=++x$ ，表示将  $x$  值赋给  $y$  之前要先将  $x$  值增加 1，然后再赋值给  $y$ ，所以  $y$  的值为 2，此时  $x$  值也为 2。然后计算  $z=x++$ ；这时先将  $x$  值赋给  $z$ ，即  $z=2$ ，然后  $x$  值再自增 1 为 3，所以程序得到的结果为  $x=3$ ， $y=2$ ， $z=2$ 。

【例题 2】下列程序的输出结果是（ ）。

```
#include "stdio.h"
void main()
{
 int a=9;
 a+=a-=a+a;
 printf("%d",a);
}
```

A. 18

B. 9

C. -18

D. -9

【解析】

选 C。本例中  $a+=a-=a+a$  要从右向左依次计算。首先计算最右侧的  $a+a$  值为 18，注意此时并未改变  $a$  的值。再计算  $a-=18$ （等价于  $a=a-18$ ），由于  $a$  原来的值为 9，因此这步运行之后  $a=-9$ 。最后计算  $a+=-9$ ，因为上步  $a$  值为 -9，所以此步等价于  $a=(-9)+(-9)$ ，所以  $a$  最终值为 -18。

【例题 3】经过下述赋值后，变量  $x$  的数据类型是（ ）。

```
float x=21.0;
int y;
```