C语言疑难问题剖析

严桂兰　刘甲耀　编著

华东化工学院出版社

内 容 提 要

本书应用当前先进的软件工程方法PAD，融软件工程与编程为一体，采用“基本内容提要→疑难问题C程序→执行结果→图文并举剖析”的方式阐述与详细剖析C语言疑难问题，内容涉及运算符、基本数据类型、包含文件、控制流、编程风格、存贮类型、指针与数组、结构与联合、预处理程序等九个方面。另外，还列出了PAD/C语言的标准程序结构图式；优先级表；运算符摘要表；对应于十进制、八进制、十六进制的ASCII码表；数据类型层次图等五个附录。

（沪）新登字208号

C语言疑难问题剖析

C Yuyan Yinan Wenti Poxi

严桂兰 刘甲耀 编著

华东化工学院出版社出版发行

（上海市梅陇路130号）

新华书店上海发行所发行经销

浙江上虞科技外文印刷厂排版

上海长鹰印刷厂印刷

开本 850×1168 1/32 印张 11 字数 293千字

1993年1月第1版 1993年1月第1次印刷

印数 1—5000册

ISBN 7-5628-0262-9/TP·22 定价 8.90元

目　录

第一章 运 算 符

C语言程序由语句所构成，语句由表达式所构成，而表达式则由运算符和操作数所构成，由于C语言的运算符既独特又丰富(详见附录三的运算符摘要表)，因此，确定运算符如何应用于操作数的规则对了解表达式有着十分重要的作用，附录二的优先级表中概述了称之为优先级和结合性的规则，我们利用该表来解答本章的问题，也就是说，严格按照运算符的优先级与结合性，把运算符与操作数结合起来，然后进行计算。当然，如果表达式中出现圆括号，则会改变运算的优先顺序。

本章介绍六个问题：

1. 基本算术运算符
2. 赋值运算符
3. 逻辑和增1、减1运算符
4. 按位运算符
5. 关系和条件运算符
6. 运算符优先级与计算

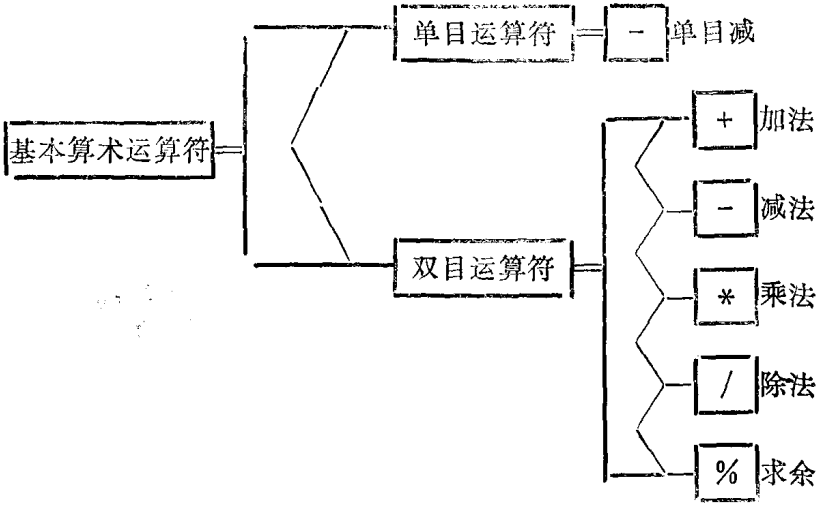
§1.1 基本算术运算符

一、基本内容提要

(一) 基本运算(见下页框图)

(二) 注意

(1) 单目运算符只有负运算符（—），其结果是操作数的负值。



(2) 除法(/)与一般算术运算规则不同,如果两个整数相除,其商取整数部分而舍去尾数,

如 5/3 = 1 (整除)

5.0/10.0 = 0.5 (实除)

(3) 模除(%)要求两个操作数均为整数,其结果为整除后取余数,即模除得整除的余数,

如 5%3 = 2; 3%5 = 3

(4) 求模运算符只用于 int 和 char 值,不能用于 float 和 double 值。

二、疑难问题

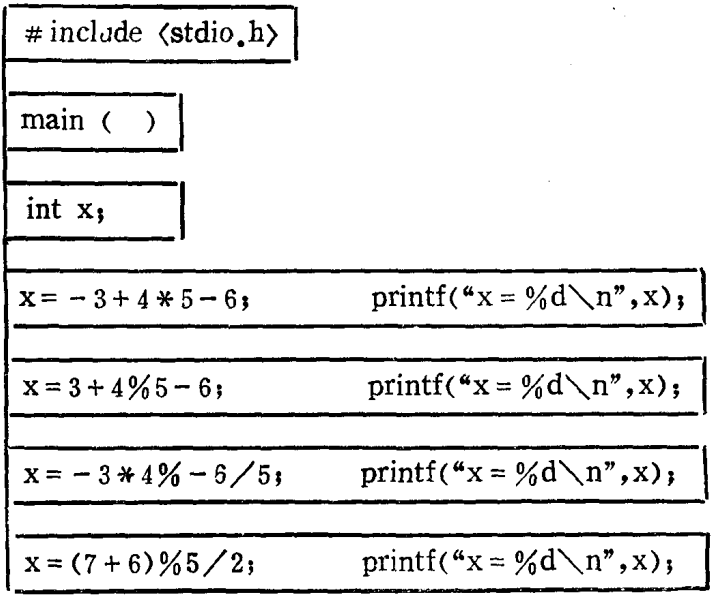
(一) 下列C程序得出什么结果

1. op1.c程序

```
# include <stdio.h>
main(    )
{
int x;
x = -3+4*5-6;     printf("x = %d\n",x);(op1.1)
```

```
x=3+4%5-6;          printf("x=%d\n",x);(op1.2)
x=-3*4%-6/5;        printf("x=%d\n",x),(op1.3)
x=(7+6)%5/2;        printf("x=%d\n",x);(op1.4)
}
```

2. C程序⇒PAD



注：有关PAD图的用法请参阅文献[1].

3. 输出结果

```
A>op 1
x=11        (op 1.1)
x=1         (op 1.2)
x=0         (op 1.3)
x=1         (op 1.4)
```

4. 剖析(见下页表)

printf是格式化打印子程序，它是标准C语言程序库的一部分。printf的第一个参数是格式串，它描述如何打印所需的任何其余的参数，对于一个参数的打印说明是以%开头。在本程序中，

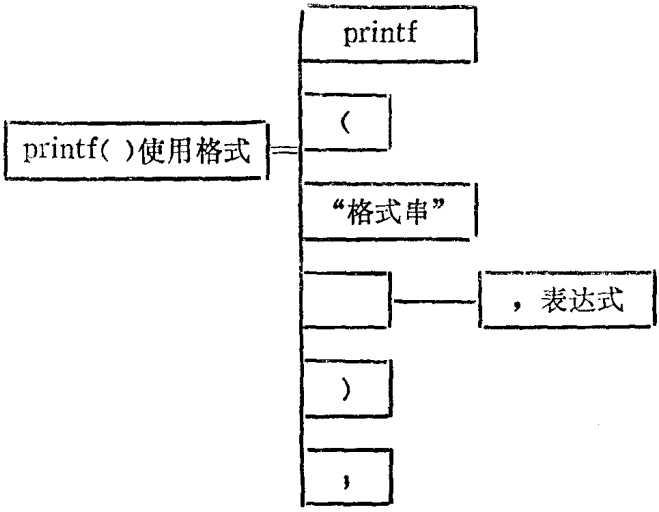
op 1.1 剖析过程	op1.1 剖析说明
x = −3+4*5−6	从附录二中了解各算术运算符优先级的高低
x = (−3)+4*5−6	在表达式中，优先级最高的运算符是单目运算符一，我们用圆括号表示操作数与运算符结合的顺序
x = (−3)+(4*5)−6	在表达式中，下一个最高优先级的运算符是*
x = ((−3)+(4*5))−6	+和−均在同一优先级，因此，结合的顺序由该级的结合性规则来决定，对+和−来说，结合性是从左到右，首先使+结合
x = (((−3)+(4*5))−6)	然后使−结合
(x = (((−3)+(4*5))−6)	最后，靠近优先级表的底部是=(优先级最低)，现在，我们已完全识别了每个运算符的操作数，因而，我们可以计算表达式之值
(x = ((−3+(4*5))−6))	对这个表达式，从内到外进行计算。
(x = ((−3+20)−6))	用它的计算结果之值替换每个子表达式
(x = (17−6)) (x = 11) 11，一个整数	赋值表达式之值是把右边的值强制转换成左边的类型

%d 告诉 printf 按十进制数解释和打印下一个参数。在后面的程序中，我们还将看到其他的打印说明，printf 也能输出文字字符。在本程序中，我们在格式串中用换行符("\n")来表示该 printf函数输出一个新行。

printf( )的使用格式可用 PAD 图描述为(见下页框图)。

即 printf("格式串"{,表达式}$_{0+}$);

其中{ }$_{0+}$ 表示重复括号中 0 次或多次。



op1.2 剖析过程	op1.2 剖析说明
x = 3 + 4%5 - 6	此表达式类似于前面的表达式
x = 3 + (4%5) - 6	按照运算符优先级与结合性进行与操作数结合
x = (3 + (4%5)) - 6 x = ((3 + (4%5)) - 6 (x = ((3 + (4%5)) - 6	模除运算符%求得4被5除的余数
(x = ((3 + 4) - 6) (x = (7 - 6)) (x = 1)	从内到外进行计算
1	最后结果

op1.3 剖析过程	op1.3 剖析说明
x = -3*4%-6/5	此表达式比前一个复杂一些，但仍按运算符的优先级和结合性规则进行计算
x = (-3)*4%(-6)/5	单目运算符，优先级最高
x = ((-3)*4)%(-6)/5 x = (((-3*4)%(-6))/5 x = ((((-3)*4)%(-6))/5) (x = ((((-3)*4)%(-6))/5)	*、%和/都是在同一优先级，而它们的结合性则是从左到右
(x = (((-3*)%(-6)/5)) (x = ((-12%-6)/5)) (x = (0/5)) (x = 0)	由内到外进行计算
0	最后结果

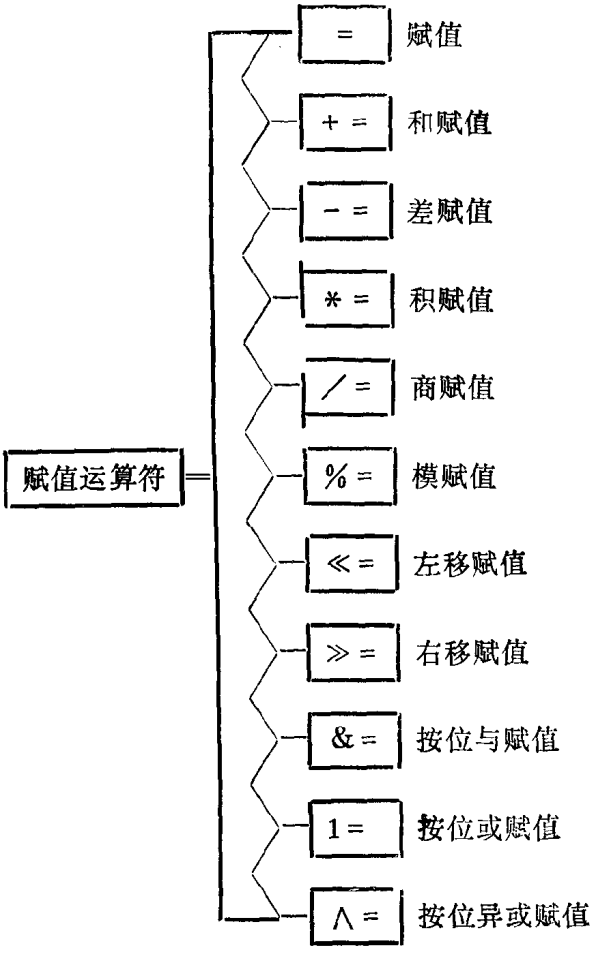
op1.4 剖析过程	op1.4 剖析说明
x = (7+6)%5/2	首先结合圆括号内的子表达式
x = ((7+6)%5)/2 x = (((7+6)%5)/2) x = (((7+6)%5)/2))	然后，如前所述那样按优先级与结合性规则进行结合
(x = ((13%5)/2)) (x = (3/2)) (x = (3/2)) (x = 1)	计算 整除运算舍去小数部分
1	最后结果

第一组疑难题的中心思想是在复杂的表达式中使用圆括号，以帮助读者把操作数与运算符结合起来，然后由内到外计算出结果。

§1.2 赋值运算符

一、基本内容提要

(一) 基本运算符



（二）注意

（1）赋值运算符左边是一个度量，是将右边表达式计算之值赋给左边变量。

（2）自反运算符是将左边变量与右边表达式进行对应计算后，再赋给左边变量。

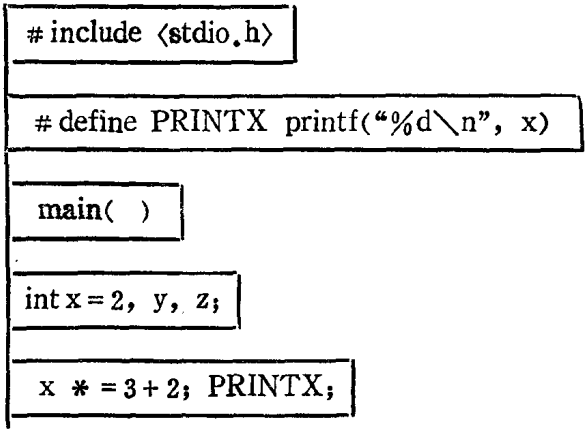
二、疑难问题

下列C程序得出什么结果

1. op2.c程序

```
#include <stdio.h>
#define PRINTX printf("%d\n",x)
main( )
{
    int x=2, y, z;
    x *=3+2;      PRINTX;          (op2.1)
    x*=y=z=4;     PRINTX;          (op2.2)
    x=y==z;       PRINTX;          (op2.3)
    x==(y=z);     PRINTX;          (op2.4)
}
```

2. C程序⇒PAD



```
x*=y=z=4;        PRINTX;

x=y==z;          PRINTX;

x==(y=z);        PRINTX;
```

3. 输出结果

```
A>op 2
10                (op 2.1)
40                (op 2.2)
1                 (op 2.3)
1                 (op 2.4);
```

4. 剖析

op 2.1 剖析过程	op 2.1 剖析说明
x=2	初始化
x*=3+2	计算顺序遵循优先级高低
x*=(3+2)	赋值运算符的优先级比算术运算低 (*=是一个赋值运算符)
(x*=(3+2)) (x*=5)	计算
(x=x*5)	改写为它的等效形式
10	最后结果

关于 define：本程序第二行为

```
#define PRINTX printf("%d\n",x)
```

在C程序中，以#开头的任一行都是C预处理程序的一个语句，由预处理程序完成的任务是用另一个字符串来替换一个字符

串。在本程序中，define语句告诉预处理程序用字符串printf("%d\n",x) 来替换字符串 PRINTX 的所有请求。这实际是宏定义方式，PRINTX 是宏， printf("%d\n",x)则是宏体

op 2.2 剖析过程	op 2.2 剖析说明
x = 10	初始化
x * = y = z = 4 x * = y = (z = 4) x * = (y = (z = 4)) (x * = (y = (z = 4)))	在此表达式中，所有运算符均赋值，因此，结合性决定其结合的顺序，赋值运算符从右到左进行结合
(x * = (y = 4)) (x * = 4)	计算
x = x * 4 = 10 * 4	等效形式
40	结果

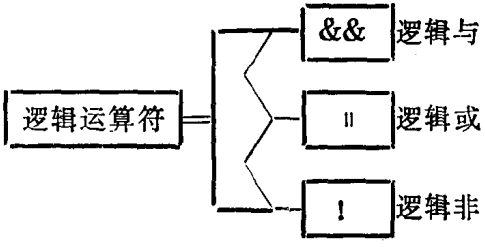
op 2.3 剖析过程	op 2.3 剖析说明
y = 4, z = 4	赋初值
x = y = = z x = (y = = z)	对C语言的一个新的编程者来说，常常把 =（赋值）和 = =（相等测试）之间的差别混淆，从优先级表可以看出，= = 在 = 之前结合
(x = (y = = z)) (x = (TRUE)) (x = 1)	相等关系运算符求得的值是逻辑值，此处为 TRUE，以整数1表示（若是FALSE则以整数 0 表示）
1	结果

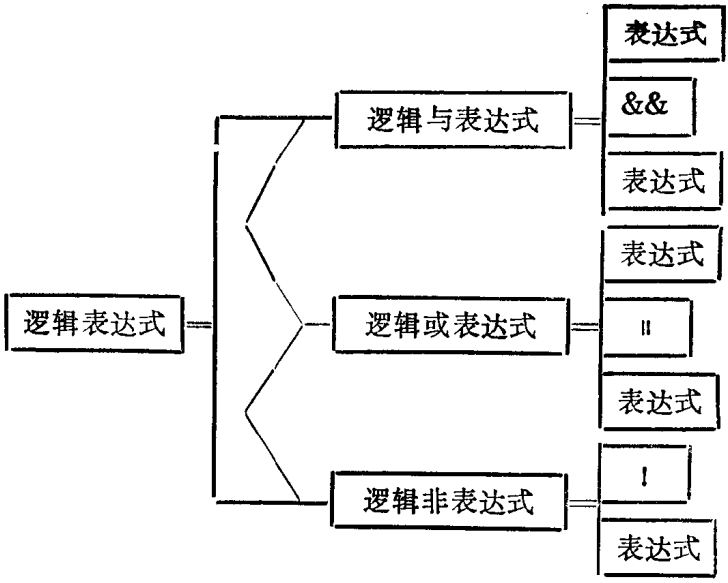
op 2.4 剖析过程	op 2.4 剖析说明
x=1,z=4	赋初值
x= =(y= z) (x= =(y=z))	在此表达式中，通过使用圆括号强迫先赋值（因相等关系运算符的优先级比赋值运算符高）
(x= =4)	计算
FALSE 或 0	x值不等于4,所以表达式之值为逻辑假或0,此时，x 的值没有改变（= =不改变它的操作数），因而，PRINTX 打印出 1

§1.3 逻辑与增1运算符

一、基本内容提要

(一) 基本运算符

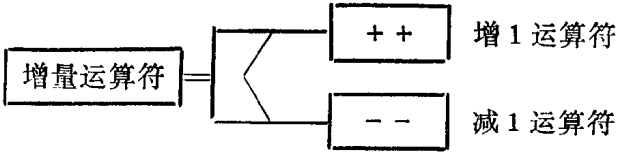


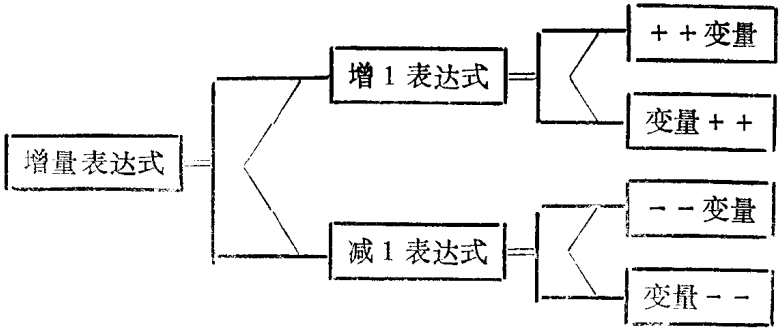


即　　逻辑与表达式::=表达式 && 表达式

逻辑或表达式::=表达式 ‖ 表达式

逻辑非表达式::=！表达式





即　　增量表达式::=++变量|变量++|--变量|变量--

（二）注意

逻辑表达式求出的值是逻辑值，真值用 TRUE 或非 0 表示，假值用FALSE 或 0 表示。

二、疑 难 问 题

（一）下列 C 程序得出什么结果？

1. OP3a.c 程序

```
#include <stdio.h>
# define PRINT(int) printf("%d\n",int)
main(    )
{
int x, y, z;
x=2; y=1; z=0;
x=x && y||z; PRINT(x);                    (op3a.1)
PRINT(x||y&&z);                           (op3a.2)
x=y=1;
z=x++-1; PRINT(x); PRINT(z);              (op3a.3)
z+=-x++ + ++y;PRINT(x); PRINT(z);         (op3a.4)
z=x/++x;        PRINT(z);                 (op3a.5)
}
```

2. C 程序⇒PAD