



第3章

数据运算

数据运算是计算机的基本功能,而数据运算主要是通过对表达式的计算完成的。C语言中运算符和表达式数据之多,在高级语言中是少见的,正是丰富的运算符和表达式使C语言功能十分完善。

3.1 运算符与表达式

数据处理是通过运算符实现的,运算符又称为操作符。为完成一个计算过程,需要使用表达式。表达式是将运算量用运算符连接起来组成的式子,其中运算量可以是常量、变量或函数。

在C语言中有很多种运算符,每一种运算符都有对应的运算规则,具体规则如下。

(1) 运算对象。根据运算对象的个数,分为一元运算符(运算对象只有一个),二元运算符(运算对象有两个),三元运算符(运算对象有3个)。

(2) 优先级。在表达式中出现多个运算符,计算表达式时必须按照一定的次序,C语言运算符都有指定的优先级,见附录A。

(3) 结合性。对于同级运算符,考虑运算的方向,就是结合性。一元运算符、三元运算符和二元的赋值(包括复合赋值)运算符均为自右向左,其余二元运算符均为自左向右。见附录A。

3.2 算术运算

用算术运算符连接数值型的运算量构成的式子,用来完成数值计算的功能。

案例 3.1 关于算术运算。

【案例描述】

通过键盘输入两个整数,分别输出两个数的相加、相减、相乘、相除和求余数的结果。

【案例分析】

在计算机上对两个整数进行运算,需要使用算术运算符。

【必备知识】

1) 算术运算符

二元运算符：加(+)、减(-)、乘(*)、除(/)、求余数(求模)(%)。

一元运算符：自增(++)、自减(--)、正负号(+、-)。

2) 运算顺序

① 括号优先；

② 一元运算符，运算顺序从右向左；

③ 二元运算符，先乘除后加减，运算顺序从左向右。

算术运算符的功能与优先顺序如表 3-1 所示。

表 3-1 算术运算符的功能与优先顺序

运算符	功 能	优先级	结合性
+ - ++ --	正号 负号 自加 自减	高	自右向左
* / %	乘 除 求余	中	自左向右
+ -	加 减	低	自左向右

3) 求余数运算(%)

两个运算对象只能是整型数据，运算结果是两个整数相除后的余数，余数也可以是负数，其符号位与被除数相同。

【案例实现】

```
#include<stdio.h>
int main( )
{
    int n1=18,n2=5;
    printf("两个数的和: n1+n2=%d\n",n1+n2);
    printf("两个数的差: n1-n2=%d\n",n1-n2);
    printf("两个数的积: n1 * n2=%d\n",n1 * n2);
    printf("两个数的商: n1/n2=%d\n",n1/n2);    //注意,两个整数相除,商取整数
    printf("两个数相除的余数=%d\n",n1%n2);
    return 0;
}
```

【运行结果】

两个数的和: n1+n2=23
两个数的差: n1-n2=13
两个数的积: n1 * n2=90
两个数的商: n1/n2=3
两个数相除的余数=3
按任意键继续...

案例 3.2 自增与自减运算。

【案例描述】

通过键盘输入一个整型变量 m=10,用如下表达式运算：

① --m

② $m++$

③ $(++m) * (m--)$

【案例分析】

自增运算符“++”和自减运算符“--”分为前置运算和后置运算,由于增减运算先后不同,从而产生运算结果不同。

【必备知识】

(1) 自增++和自减--只能对变量施加运算,不能对常量或表达式施加运算。

(2) 自增++或自减--写在变量之前,称为前缀,写在变量之后,称为后缀。例如:

$n++$ 表示先取 n 的值作为表达式的值,然后运算 $n=n+1$;

$++n$ 表示先运算 $n=n+1$,然后取 n 的值作为表达式的值。

知识拓展

变量的值与表达式的值

理解自增和自减运算符前缀和后缀的不同,最关键的是理解变量的值与表达式的值的关系,表达式为 $n++$ 或者 $++n$, 变量为 n 。后缀模式,先取值后自加,即先取变量的值作为表达式的值(此时变量为自加之前的值),然后完成变量的自加;前缀模式,先自加后取值,即先完成变量自加,然后将变量的值作为表达式的值(此时变量的值为自加之后的值)。其实如果表达式的值不作为数值参与其他运算,前缀后缀没有区别,最终只是完成了变量的自加,但是自加表达式参与其他运算,结果就有区别了。例如 $m=n++$ 和 $m=++n$, m 的结果是不同的。如果把自加运算看作是个人能力的提升过程,可以看出早提升和晚提升对于结果的影响是不同的,“只争朝夕,不负韶华”,珍惜当下,做好当下。

【案例实现】

```
#include<stdio.h>
int main()
{   int m=10;
    printf("m=%d\n",--m);           //先计算 m=m-1 的值为 9,再输出 m 的值
    printf("m=%d\n",m++);           //先输出 m 的值 9,再计算 m=m+1,值为 10
    printf("m=%d\n",m);
    printf("m=%d\n",(++m) * (m--));
    return 0;
}
```

【运行结果】

```
m=9
m=9
m=10
m=121
请按任意键继续...
```

案例 3.3 不同数据类型混合计算,系统自动进行类型转换。

【案例描述】

将不同类型的数据进行混合运算,查看最后运算结果的数据类型,采用 `sizeof()` 运算

符,测试长度,进而反推数据类型。

【案例分析】

不同数据类型的运算量混合运算时,按照一定规则进行类型转换,然后进行运算。

【必备知识】

在不同类型运算量的混合运算中,编译系统自动转换数据类型,转换的规则如下。

- ① 转换按数据精度增加方向进行,以保证数值不失真,或者精度不降低。
 - ② int 和 long 参与运算时,先把 int 类型的数据转成 long 类型后再进行运算。
 - ③ 所有的浮点型运算都是以双精度进行的,即使运算中只有 float 类型,也要先转换为 double 类型,才能进行运算。
 - ④ char 和 short 参与运算时,必须先转换成 int 类型。
 - ⑤ 运行结果的数据类型自动转换为保存运行结果变量的数据类型。
- 数据转换方向如图 3-1 所示。

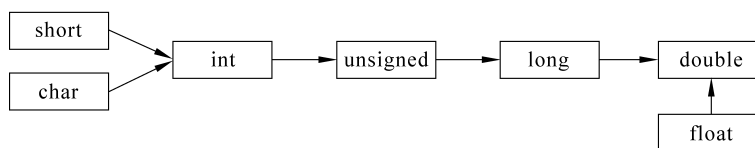


图 3-1 数据转换方向图

【案例实现】

```
#include<stdio.h>
#define PI 3.14159
int main()
{   char ch='a';
    int i=5;
    unsigned int j=6;
    long int k=12;
    float f=3.0;
    double d=6.0;
    printf("%u\n",sizeof(ch/i+i*k-(j+k)*(f*d)/(f+i)));
    return 0;
}
```

【运行结果】

```
8
请按任意键继续...
```

长度为 8 的数据类型为双精度型,因此最终输出的数据类型为双精度型。

案例 3.4 强制性数据类型转换。

【案例描述】

两个整型数据进行运算时,结果输出为浮点型。

【案例分析】

两个整型数据进行运行,无法得到浮点型结果。因此,可以采用强制类型转换完成。

【必备知识】

使用强制类型转换应注意以下几点。

- (1) 在进行强制类型转换时,类型关键字必须用“()”括住。
- (2) 在对一个表达式进行强制类型转换时,整个表达式应该用“()”括住。如果 $(float)(a+b)$ 写成 $(float)a+b$,则只对变量 a 进行了强制类型转换。
- (3) 在对变量或表达式进行了强制类型转换后,并不改变变量或表达式原来的类型。
- (4) 由数据长度长向数据长度短的强制转换时会丢掉一些数据位数,因而会造成值的改变。例如,将 $float$ 型或 $double$ 型强制转换成 int 型时,对小数部分是四舍五入还是简单地截断,取决于具体的系统。Visual C++ 采用的是截断小数的办法。

【案例实现】

```
#include<stdio.h>
int main()
{   int n1, n2;
    n1=25;
    n2=15;
    printf("n1/n2 结果为浮点数值: %f\n", (float)n1/n2);
    return 0;
}
```

【运行结果】

n1/n2 结果为浮点数值: 1.666667
请按任意键继续...

3.3 赋值运算

在 C 语言中,赋值运算符“=”与赋值语句紧密相关,赋值表达式的末尾加上“;”就是赋值语句。

案例 3.5 当整形变量 a 初值 12,完成以下规定的表达式运算,输出其结果。表达式: $a+=a-=a*a$ 。

【案例描述】

复合赋值运算符,左侧为对其赋值的变量,后为运算式,其运算结果就是更新前面的变量值,加深理解运算的结合性:自右向左。

【案例分析】

在计算赋值表达式 $a+=a-=a*a$ 时,按从右到左的顺序先计算 $a*a$,得 144,然后进行 $a-=144$ 的赋值运算,得到 $a=12-144=-132$,该值作为赋值表达式 $a-=a*a$ 的值,参加下一个赋值表达式的计算,得 $a=a+(-132)=-264$ 。因此,当同一变量被连续赋值时,要注意其值在不断变化。

【必备知识】

① “=”在 C 语言中不是表示相等关系,而是进行赋值运算,即将“=”右侧的值赋给“=”左侧的变量。

② “+=、-=、*=、/=、%=”是复合赋值运算符,具有算术运算和赋值的双重功

能。在使用复合赋值运算符时,要将右边的表达式作为一个整体与左边的变量进行运算。

③ 赋值运算符的优先级是比较低的,结合性均为从右到左。参见附录 A。

知识拓展

赋值运算符的启示

在 $v=e$ 这样的赋值表达式中,当 v 和 e 的数据类型不一致时,C 编译系统自动将 e 的类型转换成与 v 相同的类型后再赋值。

e 本身有自己的数据类型,但是当它位于赋值号右侧时,其数据类型已经不重要了,最终赋值的类型由 v 决定,这是由二者的所处的位置决定的。赋值符号如此,人生的很多问题亦是如此,只不过人生中的位置更多指的是每个人努力后的内在实力。

【案例实现】

```
#include <stdio.h>
int main()
{   int a=12;
    a+=a-=a*a;          //该表达式可拆分为 a=a-a*a;   a=a+a;
    printf("%d\n",a);
    return 0;
}
```

【运行结果】

-264

3.4 逗号运算(顺序运算)

逗号运算符“,”的作用是连接多个数据项,从而将它们作为一个整体来处理。

案例 3.6 计算下面逗号表达式的值,并对比输出结果。表达式为 $a=3*5,a*4,a+5$ 和 $a=(3*5,a*4,a+5)$ 。

【案例描述】

逗号表达式,是将多个表达式串在一起,顺序计算各个表达式的值。将最后一个表达式的值作为整个表达式的值。

【案例分析】

在计算逗号表达式 $a=3*5,a*4,a+5$ 中,按表达式顺序,先计算 $a=3*5=15$;再计算 $a*4=15*4=60$;最后计算 $a+5=15+5=20$,结果 20 作为逗号表达式的值。

【必备知识】

(1) 逗号运算符在 C 语言中优先级是最低的,参见附录 A。

(2) 结合性: 从左到右。

(3) 逗号表达式的结果: 最后一个表达式的值。

(4) 逗号表达式应用。

① 用一个逗号表达式语句可代替多个赋值语句。

```
a=0;b=1;c=2;
```

可写成

```
a=0,b=1,c=2;
```

② 用一个逗号表达式语句可得到多个计算结果。

```
y=10;  
x=(y=y-5,60/y);
```

执行后,x 的值为 12,y 的值为 5。

注意以下两个语句的区别:

```
a=3*5,a*4;           /* 逗号表达式的值为 60 */  
x=(a=3,3*6);          /* 有赋值运算符,将右侧的逗号表达式的值赋给变量 x,  
                        /* "()" 中逗号表达式值为 18
```

③ 当某些语法位置只允许出现一个表达式时,用逗号表达式可实现多个表达式的运算,例如循环中 for 语句:

```
for(i=0,j=0;i<8,j<10;i++,j++)
```

【案例实现】

```
#include <stdio.h>  
int main()  
{   int a=5;  
    printf("%d\n", (a=3*5,a*4,a+5));  
    a=5  
    printf("%d\n", a=(3*5,a*4,a+5));  
    return 0;  
}
```

【运行结果】

```
20  
10  
请按任意键继续...
```

3.5 关系运算和逻辑运算

在程序设计中,关系运算和逻辑运算通常用在程序流程控制中的 if、switch、for 或 while 语句中,目的是进行条件判断,以便程序能按照指定的流程运行。

3.5.1 关系运算

案例 3.7 输入两个整数,找出其中较大的值。

【案例描述】

通过键盘输入两个数,进行比较,找出值较大的数。

【案例分析】

要进行两个数的比较,需要用到关系运算符和分支语句。

【必备知识】

关系就是将两个数据进行比较,判定两个数据是否符合给定的关系。关系运算符是用来比较两个数据的运算符,运算结果为逻辑值:真和假(分别用整数 1 和 0 表示),表 3-2 列出 C 语言中的关系运算符和用法。

表 3-2 关系运算符和用法

运算符	名 称	优先级	范 例	结 果	结合性
>	大于	6	$3>2$	1	自左向右
>=	大于或等于		$3>=2$	1	
<	小于		$3<2$	0	
<=	小于或等于		$3<=2$	0	
==	等于	7	$3==2$	0	
!=	不等于		$3!=2$	1	

注意:

- ① 关系运算符“==”不要误写成赋值运算符“=”。
- ② 优先级数值越大,参与运算的顺序靠后。

知识拓展

如何表述数据范围

一个关系运算符可以表述数据之间的关系,如果想表示数据范围呢?数学中常见的 $m \leq x \leq n$ 的形式是否就是区间 $[m, n]$ 正确的计算机表述呢?可以按照关系运算符的从左向右的规则计算一下上述形式是否正确,结果显而易见, $m \leq x$ 的运算结果只有两种可能性,“真”或者是“假”,其数值为“1”或者“0”,显然 $0 \leq n$ 或者 $1 \leq n$ 并不是该表达式想要表述的含义,这时就需要用到逻辑运算符。

【案例实现】

```
#include<stdio.h>
int main()
{   int x,y,max;
    printf("请输入两个整数: ");
    scanf("%d,%d",&x,&y);
    if (x>y)
        max=x;
    else
        max=y;
    printf("两个整数中较大值=%d\n",max);
    return 0;
}
```

【运行结果】

请输入两个整数: 5 7

两个整数中较大值=7
请按任意键继续...

3.5.2 逻辑运算

案例 3.8 计算逻辑表达式中变量 k 、 x 、 y 、 z 的值。表达式为 $k = x + + > = 0 \& \& !(y - - < = 0) \parallel (z = x + y)$ 。

【案例描述】

在复杂表达式运算中,掌握运算符的优先级和逻辑运算规定的运算规则。

【案例分析】

逻辑运算符使用 3 个时,按照逻辑运算符的优先级和结合性进行运算。

【必备知识】

1) 逻辑运算符

逻辑运算符用来对两个关系式或逻辑量进行逻辑运算,运算结果为逻辑值:真和假(分别用整数 1 和 0 表示),表 3-3 列出 C 语言中的逻辑运算符和用法。

表 3-3 逻辑运算符和用法

运 算 符	名 称	优 先 级	结 合 性
!	逻辑非	2	自右向左
&.&	逻辑与	11	自左向右
	逻辑或	12	

2) 逻辑运算规则

参加逻辑运算的运算量可以是任何类型的数据。在 C 语言中进行逻辑运算时,由于 C 语言本没有逻辑类型,在内部计算中使用整数表达关系运算和逻辑运算的结果,0 表示逻辑假,而非 0 的值表示逻辑真,非 0 值表示逻辑真,0 值表示逻辑假。表 3-4 为逻辑运算真值表。

表 3-4 逻辑运算真值表

a	b	a & b	a b	!a
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

 知识拓展

如何理解 C 语言的逻辑值? 为了方便记忆,总结如下。整数 0 为逻辑假,非 0 都是逻辑真;逻辑真的值是 1,逻辑假的值是 0。

注意: 逻辑运算符出现在不同表达式中时其计算方法和顺序不同,具体如下。

① 当逻辑运算符 &.&,在计算逻辑表达式,如(表达式 1)&.&(表达式 2)&.&...时,

表达式 1 的值为“假”，后面表达式不再进行计算，整个逻辑表达式的值为“假”。

② 当逻辑运算符 `||`，在计算逻辑表达式，如 (表达式 1) `||` (表达式 2) `||` ... 时，表达式 1 的值为“真”，后面表达式不再进行计算，整个逻辑表达式的值为“真”。

③ 当 3 个运算符有两个或两个以上同时使用时，根据运算符的优先级和结合性进行运算。

【案例实现】

```
#include<stdio.h>
int main()
{   int x=-1, y=5, z=6, k;
    k=x++>=0&&!(y--<=0)|| (z=x+y);           //x++>=0 为假, y--<=0 不计算
    printf("k=%d, x=%d, y=%d, z=%d\n", k, x, y, z);
    return 0;
}
```

【运行结果】

```
k=1, x=0, y=5, z=5
请按任意键继续...
```

知识拓展

数据范围表述

内容学习到这里，应该已经理解了如何用计算机表达式来表述数据范围了，那就是用逻辑与，区间 $[m, n]$ 正确的计算机表达式为 $m \leq x \ \&\& \ x \leq n$ 。所以学习知识切忌生搬硬套，刻舟求剑是不可取的。

3.6 位运算

C 语言和其他语言不同的是，它完全支持按位运算，主要用于编写系统软件，实现汇编语言能够对位操作的功能。

案例 3.9 对十进制整数 25 和 120，分别进行位与，位或，左移 2 位，右移 2 位和位取反操作。

【案例描述】

定义两个整型变量 $a=25, b=120$ ，进行位运算的各种操作后，显示变量 a, b 值发生的变化。

【案例分析】

十进制数转换成二进制数，对二进制数进行位与，位或，位异或，移位和位取反操作，每个运算量的二进制位上的 0 和 1 会发生相应变化。

【必备知识】

1) 位运算符

位运算符是针对二进制数的每个二进制位进行运算的符号，对数字 0 和 1 进行二进制位操作，结果如表 3-5 所示。

表 3-5 二进制位运算符

运 算 符	名 称	优先级	范 例	结 果	结合性
&	按位与	8	0&0	0	自左向右
			0&1	0	
			1&0	0	
			1&1	1	
	按位或	9	0 0	0	自左向右
			0 1	1	
			1 0	1	
			1 1	1	
^	按位异或	10	0^0	0	自左向右
			0^1	1	
			1^0	1	
			1^1	0	
~	按位取反	2	~0	1	自右向左
			~1	0	
<<	左移位	5	00000010<<2	00001000	自左向右
			10010011<<2	01001100	
>>	右移位		01100010>>2	00011000	自左向右
			11100010>>2	11111000	
&=、 =、^=、<<=、>>=	位复合赋值运算符	14	略	略	自右向左

2) 位操作符

位操作符分为两类,一类为位逻辑运算,另一类为移位运算。位逻辑运算实现二进制位与二进制位之间的逻辑运算,移位运算实现二进制位的顺序向左或向右移位。

注意:

- (1) 位逻辑运算的运算量只能是字符型和整型数据。
- (2) 与逻辑运算的结果只能是 0 或 1 不同,按位逻辑运算的结果可能是任何整数值。

3) 位运算 6 种操作

(1) 按位与(&)。例如:

```
int a=65,b=120,c;      //a,b 看成 1B 长度的整型数
c=a&b;
```

```

      0 1 0 0 0 0 1
&  0 1 1 1 1 0 0
-----
      0 1 0 0 0 0 0
```

(结果:十进制整数 64)

(2) 按位或(|)。例如：

```
int a=65,b=120,c;      //a,b 看成 1B 长度的整型数
c=a|b;
```

```

      0 1 0 0 0 0 0 1
    | 0 1 1 1 1 0 0 0
    -----
      0 1 1 1 1 0 0 1    (结果：十进制整数 121)
```

(3) 按位异或(^)。例如：

```
int a=65,b=120,c;      //a,b 看成 1B 长度的整型数
c=a^b;
```

```

      0 1 0 0 0 0 0 1
    ^ 0 1 1 1 1 0 0 0
    -----
      0 0 1 1 1 0 0 1    (结果：十进制整数 57)
```

(4) 按位取反(~)。例如：

```
int a=65,b;            //a 看成 1B 长度的整型数
b=~a;
```

```

    ~ 0 1 0 0 0 0 0 1
      1 0 1 1 1 1 1 0    (结果：十进制整数 190 或 -66)
```

(5) 左移位(<<)。例如：

```
int a=25,b;
b=a<<3;
```

对无符号数,左移位相当于乘 2 运算,左移 n 位相当于该数乘 2^n ,在本例中, $25 \ll 3$ 表示 $25 \times 2^3 = 200$,如图 3-2 所示。



图 3-2 将 a 左移 3 位存入 b

(6) 右移位(>>)。在移位过程中,各个二进制位顺序向右移动,移出右端之外的位被舍弃,左端空出的位是补 0 还是补 1 取决于具体的机器和被移位的数是有符号数还是无符号数。具体如下。

① 对无符号数进行右移时,左端空出的位一律补 0。

② 对用补码表示的有符号数进行右移时,有的机器采取逻辑右移,有的机器则采取算术右移。逻辑右移时,不管是正数还是负数,左端空位一律补 0。算术右移时,正数右移,左端的空位全部补 0;负数右移,左端的空位全部补 1(即符号位)。Visual C++ 2010 采用的是算术右移。例如：

```
int a=-32768,b;
b=a>>2;      //逻辑右移和算术右移得到不同的值
```

① 逻辑右移: a 的各个二进制位顺序右移 2 位,其值为 8192。

② 算术右移,左侧空格补 1,值为 -8192。

采用算术右移相当于除 2 运算。右移 1 位,其值等于该数除以 2,右移 n 位,其值等于该数除以 2^n 。本例中, $-32768 >> 2$ 表示 $-32768 \div 2^2 = -8192$,如图 3-3 所示。

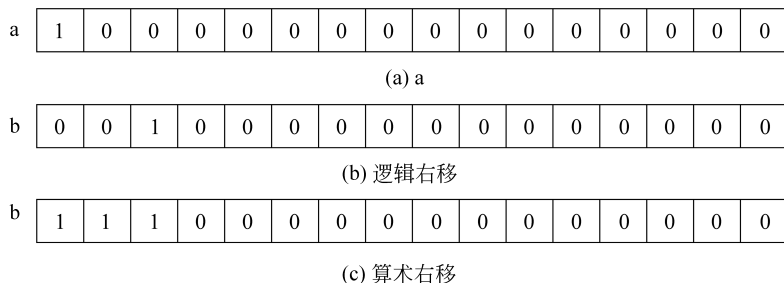


图 3-3 将 a 进行右移存入 b

【案例实现】

```
#include <stdio.h>
int main()
{   int a=25,b=120,c1,c2,n=1;
    c1=a,c2=b;           //c1,c2 为 a,b 的原值
    a&=b;
    b|=a;
    printf("a=%d&%d=%d   b=%d|%d=%d\n",c1,c2,a,c2,c1,b);
    a=c1,b=c2;
    a>>=n+1;
    b<<=n+1;
    printf("a=%d>>=%d=%d   b=%d<<=%d=%d\n",c1,n+1,a,c2,n+1,b);
    a=c1,b=c2;
    a=~ a;
    b^=a;
    printf("a=~ %d=%d   b=%d^%d=%d\n",c1,a,c2,c1,b);
    return 0;
}
```

【运行结果】

```
a=25&20=24   b=120|25=120
a=25>>2=6    b=120<<2=480
a=~25=-26    b=120^25=-98
请按任意键继续...
```

知识拓展

运算符的优先级

- ① 括号的优先级最高;
- ② 单目(一元)运算符高于双目运算符;
- ③ 算术运算高于关系运算;

- ④ 关系运算高于逻辑运算；
- ⑤ 逻辑运算高于赋值运算；
- ⑥ 逗号运算最低。

习题 3

一、选择题

- C 语言中要求运算量必须是整型的运算符是()。
A. + B. / C. % D. -
- 在 C 语言中,不同类型的数据混合运算时,先要转换成同一类型,然后进行计算。设一表达式中含有 int、long、unsigned 和 char 类型的常数和变量,则表达式的最后运算结果是(【1】)类型。这四种类型的转换规律是(【2】)。
【1】A. int B. char C. unsigned D. long
【2】A. int→unsigned→long→char B. char→int→long→unsigned
C. char→int→unsigned→long D. char→unsigned→long→int
- a、b 均为整数,且 $b \neq 0$,则表达式 $a/b * b + a \% b$ 的值为()。
A. a B. b
C. a 被 b 除,结果的整数部分 D. a 被 b 除,结果的整数部分
- a、b 均为整数,且 $b \neq 0$,则表达式 $a - a/b * b$ 的值为()。
A. 0 B. a
C. a 被 b 除的余数部分 D. a 被 b 除,商的整数部分
- 表达式 $18/4 * \text{sqrt}(4.0)/8$ 的数据类型为()。
A. int B. float C. double D. 不确定
- 若变量已正确定义,且 k 的值是 4,执行表达式 $j = k--$ 后,j、k 的值是()。
A. $j=4, k=4$ B. $j=4, k=3$
C. $j=3, k=4$ D. $j=3, k=3$
- 设 $\text{int } x=10, x+=3+x\%(-3)$,则 $x=()$ 。
A. 14 B. 15 C. 11 D. 12
- 表达式 $(\text{int})(3.0/2.0)$ 的值是()。
A. 1.5 B. 1.0 C. 1 D. 0
- 设 $\text{float } m=4.0, n=4.0$; 使 m 为 10.0 的表达式是()。
A. $m-=n*2.5$ B. $m/=n+9$ C. $m*=n-6$ D. $m+=n+2$
- C 语句 $x*=y+2$;还可以写成()。
A. $x=x*y+2$; B. $x=2+y*x$;
C. $x=x*(y+2)$; D. $x=y+2*x$;
- 若变量已正确定义,要将 a 和 b 中的数进行交换,则下列不正确的语句组是()。

- A. $a=a+b, b=a-b, a=a-b;$ B. $t=a, a=b, b=t;$
 C. $a=t; t=b; b=a;$ D. $t=b; b=a; a=t;$
12. 设已定义: $\text{int } k=7, x=12;$ 下列表达式中, 计算结果为 0 的是()。
 A. $x\%=(k\%=5)$ B. $x\%=(k-k\%5)$
 C. $x\%=k-k\%5$ D. $(x\%=k)-(k\%=5)$
13. 若已定义 x 和 y 为 double 型变量, 则表达式: $x=1, y=x+3/2$ 的值是()。
 A. 1 B. 2 C. 2.0 D. 2.5
14. 设 $\text{int } c=5$ 和 $\text{int } a, a=2+(c+=c++, c+8, ++c);$ 则 a 的值为()。
 A. 15 B. 14 C. 13 D. 16
15. 设 $\text{int } a=7, b=8;$ 则 $\text{printf}("%d, \%d", (a+b, a), (b, a+b));$ 的输出是()。
 A. 7, 15 B. 8, 15 C. 15, 7 D. 出错
16. 设 $\text{int } a=3;$ 则表达式 $a<1\&\&--a>1$ 的运算结果和 a 的值分别是()。
 A. 0 和 2 B. 0 和 3 C. 1 和 2 D. 1 和 3
17. 已知 $\text{int } x=43, y=0; \text{char } ch='A';$ 则表达式 $(x>=y\&\&ch<'B'\&\&!y)$ 的值是()。
 A. 0 B. 语法错 C. 1 D. -1
18. 为表示“ a 和 b 都大于 0”, 应使用的 C 语言表达式是()。
 A. $(a>0)|(b>0)$ B. $a\&\&b$
 C. $(a>0)|| (b>0)$ D. $(a>0)\&\&(b>0)$
19. 设 a 为整型变量, 下列不能正确表达数学关系: $10<a<15$ 的 C 语言表达式是()。
 A. $10<a<15$
 B. $a==11||a==12||a==13||a==14$
 C. $a>10\&\&a<15$
 D. $!(a<=10)\&\&!(a>=15)$
20. 设有以下语句, 则 z 的二进制值是()。
 $\text{char } x=3, y=6, z; z=(x^y)<<2;$
 A. 00010100 B. 00011011 C. 00011100 D. 00011000

二、填空题

1. 设有以下定义, 并已赋确定的值:

```
char ch; int i; float f; double d;
```

则表达式 $ch*i+d-f$ 的数据类型为_____。

2. 设有 $\text{int } a=11;$, 则表达式 $(a++*1/5)$ 的值为_____。

3. 运行以下程序段, 若要使 $a=5.0, b=4, c=3$, 则输入数据的形式应为_____。

```
int b, c;
float a;
```

```
scanf("%f,%d,c=%d",&a,&b,&c);
```

4. 设已定义 `int a=10,b=12`, 则表达式 `!a||b--` 的值是_____。

5. 若 `a` 为 `int` 型变量, 请以最简单的形式写出与逻辑表达式 `!a` 等价的 C 语言关系表达式_____。

6. 若有以下程序段

```
int a=1,b=2,c;  
c=1.0/b*a;
```

则执行后, 变量 `c` 的值为_____。

三、编程题

1. 编写程序, 其功能是从键盘输入长方体三边的边长 x 、 y 、 z , 然后求其表面积 a 和体积 v 。

2. 编写程序, 其功能是从键盘输入圆的半径 r , 计算并输出该圆的面积 s 和周长 c 。

3. 编制程序, 其功能是从键盘输入 3 个顶点的坐标 (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , 按海伦公式计算三角形的面积 s 。