

C语言 03 运算符和表达式

运算符

运算符

基本知识

算术运算符

- + * / % (取余数)

关系运算符

>=, >, <, <=, != (不等于), == (等于)

逻辑运算符

! (非), && (并且), || (或者)

赋值运算符

=, +=, *=, /=, -=

如, a +=1; 等价于 a = a+1;

优先级

算数 > 关系 > 逻辑 > 赋值

除法与取余

除法的运算结果和运算对象的数据类型有关

都是int——》结果只取整数部分

有一个是float——》商也是浮点型，保留小数

取余的对象必须是整数

结果的符号与被除数相同

13%3 == 1;

-13%3 ==-1;

逻辑运算符

!真

假

!假

真

真&&真

真

真&&假

假

假&&真

假

假&&假

假

真||假

真

假||真

真

真||真

真

假||假

假

以下三种情况，m和k的值分别是多少？

```
int k = 20;
```

1. `m = (3>2) && (k=5);` ——》 1, 5

2. `m = (3>2) && (k=0);` ——》 0, 0

3. `m = (1>2) && (k=5);` ——》 0, 20

4. `m = (1>2) && (k=0);` ——》 0, 20

```
printf("m = %d, k = %d", m, k);
```

心得：

&&前面为真，就执行后面的操作，即给k赋值，但把0赋给k后，根据C语言判断真假的特性（1为真，0为假），判定k=0为假

&&前面为假，就不再执行后面的语句

一些琐碎的运算符知识

自增和自减

分类：

前自增减 `-- ++i`

后自增减 `-- i++`

相同

都是表达 $i + 1$ 后的值

不同

前自增整体表达式的值是 $i+1$ 之后的值

后自增整体表达式的值是 $i+1$ 之前的值

```
i = j = 3;
k = i++;
m = ++j;

Microsoft Visual S
后自增 i = 4
前自增 j = 4
后自增 k = 3
前自增 m = 4
```

注意

1.别管这两个的区别

2.自增/自减应单独成一个语句，不要把它放在完整的大语句中使用。

如： `int m = i++ + ++i + i + i++;`

三目运算符

`A ? B : C`

等价于

`if (A)`

`语句B;`

`else`

`语句C;`

逗号表达式

(A, B, C, D)

功能：

从左到右执行

最终表达式的值是最后一项的值