自增运算符“++”与自减运算符“--”都是单目运算符的一种。根据它们所处操作数的位置可以分为前缀和后缀形式，这两种形式在某些情况下产生的效果是一毛一样的。但我们更关心是它们的不同之处，下来就看看它们不同之处的本质区别在哪！

    前缀形式的++操作符出现在操作数的前面。

    例如:   ++a;

    说明: 操作数的值被相加，而表达式的值就是操作数****增加后的值****。

    举例:   int a, b = 1;

            a = ++b;        //b增加至2，a得到的 值为 2

    后缀形式的++操作符出现在操作数的后面。

    例如:   a++;

    说明: 操作数的值仍被相加，而表达式的值是操作数****增加前的值****。

    举例:   int a, b = 1;

            a = b++;        //b增加至2，a得到的值为1

    显然，根据操作符的位置，我们很容易记住这个规则:

    在操作数之前的操作符(++b)在变量值在被使用之前增加它的值；在操作数之后的操作符(b++)在变量值被使用之后才增加的值。--操作符的工作原理与此相同，只是它所执行的是减值操作而非增值操作。

     在这里，有些大虾可能就要吐槽了: 说的这些都是老套的东西，我都能背下来了！能不能说些新的、我们不知道知识？

           行！行！行！先别喷，我说一个我也是才知道的，大虾可能早就见过了！但我想，我这可能还没那么多的大虾 ^\_^  更多的是和我一样的小菜，那我们一起来看看!

    int a = 1;

    ++a = 5;

    首先说明！上面第二行的表达式是有问题的，我来解释一下为什么有问题。

    划重点！

    抽象地说，前缀和后缀形式的增值操作符都复制了一份变量值的拷贝。用于周围表达式的值正是这份拷贝(在上边的例子中，“周围表达式”是指赋值操作)。前缀操作符在进行复制之前增加变量的值，后缀操作符在进行复制之后才增加变量的值。这些操作符的结果不是被它们所修改的变量，而是变量值的拷贝，认识到这一点很重要！

    我们再看这个表达式的错误之处:

    ++a = 5;

    ++a 的结果是a值的拷贝，并不是变量本身，也就是它的结果是一个常量值，你无法向一个值进行赋值。

