



知识点多总结





- 复合运算符是一种将算术运算和赋值运算组合在一起的运算符，在执行运算后，会把结果值赋给左边的变量。

- 常见的有：

加法赋值运算符：`+=`

减法赋值运算符：`-=`

乘法赋值运算符：`*=`

除法赋值运算符：`/=`

取余赋值运算符：`%=`

按位与赋值运算符：`&=`

按位或赋值运算符：`|=`

按位异或赋值运算符：`^=`

左移赋值运算符：`<<=`

右移赋值运算符：`>>=`



- 使用示例：`a += b;` (等价于 `a = a + b;`)

注意：

- 1) `a`是变量，`b`可以是具体值，也可以是变量，还可以是一个式子。
- 2) 无论`b`是哪种形式，在执行运算时，通常会先计算出右边`b`的值，再跟左边的`a`，按照复合运算符进行相应的运算，并把结果值赋回给`a`。



· **自增运算符 (++) 和自减运算符 (--) 只能跟变量结合使用，表示把变量的值加1或减1。**

例子：`i++;`(√) `5++;` (×)

· **自增、自减运算符可以放在变量的前面或后面。**

1) **前置形式 (放在变量前面)：**会先给变量加1或减1，再取变量修改后的值，参与后续操作。**(先加减，再取值。)**

比如：假设有变量i和j，i的初始值为2。

`j = ++i;` // 先给i加1，再取i修改后的值，并赋给j。执行后， $i = 2 + 1 = 3$, $j = i = 3$ 。

`j = --i;` // 先给i减1，再取i修改后的值，并赋给j。执行后， $i = 2 - 1 = 1$, $j = i = 1$ 。

2) **后置形式 (放在变量后面)：**会先取变量原本的值，参与后续操作，再给变量加1或减1。**(先取值，再加减。)**

比如：假设有变量i和j，i的初始值为2。

`j = i++;` // 先取i原本的值2，并赋给j，再给i加1。执行后， $j = i = 2$, $i = i + 1 = 3$ 。

`j = i--;` // 先取i原本的值2，并赋给j，再给i减1。执行后， $j = i = 2$, $i = i - 1 = 2$ 。



- 表达式和语句：

`1 + 2 * a - 3` ==> 是表达式。

`count = 1 + 2 * a - 3;` ==> 是语句。

· **表达式**是由一个或多个运算对象（具体值、变量等等），与零个或多个运算符组合成的式子。表达式是任何可以计算出值的代码片段。字面值和变量是最简单的表达式。

· **语句**不返回值，而是用来完成某些操作或控制执行流程的。



题目1、表达式 $a += a -= a = 9$ 的值是 (D)

- A. 9
- B. -9
- C. 18
- D. 0



题目2、有以下程序

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

程序运行后的输出结果是 (A)

- A. -12
- B. 9
- C. 0
- D. 3



题目3、有以下程序

```
#include<stdio.h>
main()
{
    int x=010,y=10;
    printf("%d, %d\n",++x,y--);
}
```

程序运行后的输出结果是 (D)

- A. 10, 9
- B. 11, 10
- C. 010, 9
- D. 9, 10



题目4、有以下程序：

```
#include <stdio.h>
main()
{
    int i = 1;
    int sum = 1;

    sum += i++;
    printf("%d\n", sum);
}
```

程序的输出结果是 (A)。

- A. 2
- B. 1
- C. 4
- D. 3



题目5、设有定义： int x = 2;

以下表达式语句中，值不为6的是 (D)。

- A. $x *= x + 1;$
- B. $x++; x *= 2;$
- C. $x *= (1 + x);$
- D. $x += x++ * 2;$