

The cryptic C language shortcut for this operation is

```
i += 5
```

This line means “Increase the value of variable *i* by five.” Unfortunately, it just doesn’t *look* like that’s what it means.

Although I can swallow `++` to increment and `--` to decrement, the `+=` thing seriously looks like a typo. Sad news: It’s not. Even sadder: There are more of them, one each for adding, subtracting, multiplying, or dividing a variable’s value by a certain amount (or by another variable). Table 16-2 lists the lot of them.

<i>Long, Boring Way</i>	<i>Cryptic Shortcut</i>
<code>var=var+5</code>	<code>var+=5</code>
<code>x=x+y</code>	<code>x+=y</code>
<code>var=var-5</code>	<code>var-=5</code>
<code>x=x-y</code>	<code>x-=y</code>
<code>var=var*5</code>	<code>var*=5</code>
<code>x=x*y</code>	<code>x*=y</code>
<code>var=var/5</code>	<code>var/=5</code>
<code>x=x/y</code>	<code>x/=y</code>

In Table 16-2, you see two examples for each cryptic shortcut. The first one uses the variable *var*, which is modified by a constant value, 5. The second uses two variables; the first one, *x*, is modified by another variable, *y*.

Yes, the shortcuts for incrementing, decrementing, or changing a variable are cryptic. You don’t have to use them. You suffer no penalty for forgetting about them. I refer to them here for two reasons: It can be done, and C gurus love tossing this stuff into programs; so don’t let the shortcuts scare you when you see them.

On your own: Modify the preceding two programs, CHANT.C and 1000.C. Replace the long math condition in the `for` loop with a shortcut version. Answers are provided at the end of this chapter.