Silly formatting trivia

The `if-else` structure need not be heavy-laden with curly braces. Just as you can abbreviate an `if` statement to one line, you can also abbreviate `else`. I don't recommend it, which is why I'm terribly brief and don't ever show a program that illustrates examples this crudely:

```
if(tax1>tax2)
{
    printf("You owe $%i in
   taxes.\n",tax1*10);
}
else
{
    printf("You owe $%i in
   taxes.\n",tax2*10);
}
```

In this example, you see the meat and potatoes of the TAXES.C program: the `if-else` structure. Because both `if` and `else` have only one statement belonging to them, you can abbreviate the source code this way:

```
if(tax1>tax2)
    printf("You owe $%i in
   taxes.\n",tax1*10);
else
    printf("You owe $%i in
   taxes.\n",tax2*10);
```

This format keeps the indenting intact, which is one way to see what belongs to what (and also to easily identify the `if-else` structure). The following format is also possible, though it makes the program hard to read:

```
if(tax1>tax2) printf("You owe
   $%i in taxes.\n",tax1*10);
else printf("You owe $%i in
   taxes.\n",tax2*10);
```

Everything is scrunched up on two lines; the `if` statement has its own line, and the `else` has its own line. Both lines end with a semicolon, which is how this works as two statements in the C language. But, look-it. It's gross! Please don't write your programs this way.

You can do this trick — eliminating the curly braces — whenever only one statement appears with an `if` or `else` keyword. If multiple statements must be executed, you're required by law to use the curly braces. That's why I recommend them all the time: No sense risking prison over brevity. To wit:

```
if(tax1>tax2)
    printf("You owe $%i in
   taxes.\n",tax1*10);
else
{
    printf("You owe $%i in
   taxes.\n",tax2*10);
    printf("It pays to live
   where it's cold!\n");
}
```

Because two `printf` statements belong to the preceding `else`, the curly braces are required.

Either-or conditions are the daily bread of the `if-else` duo. Either way, one set of statements is executed and not the other, depending on the comparison made by `if`.

What about "one, two, or the third" types of decisions? For them, you need the miraculous and overly versatile `else-if` combination. It really drives you batty, but it's handy.