Save the file back to disk.

Compile TAXES.C. Run the final result. The output is the same because the program hasn't changed (and assuming that it hasn't gotten any warmer and you haven't grown any taller in the past few moments). What you have done is to create an `if-else` *structure,* which is another way to handle the decision-making process in your C programs.

- The `else` keyword is a second, optional part of an `if` cluster of statements. It groups together statements that are to be executed when the condition that `if` tests for isn't true.
- Or else what?
- Alas, if you enter the same values as in the old program, you still owe the same bundle to Uncle Sam.

Covering all the possibilities with `else`

The `if-else` keyword combination allows you to write a program that can make either-or decisions. By itself, the `if` keyword can handle minor decisions and execute special instructions if the conditions are just so. But when `if` is coupled with `else`, your program takes one of two directions, depending on the comparison `if` makes. Figure 12-2 illustrates how this can happen.

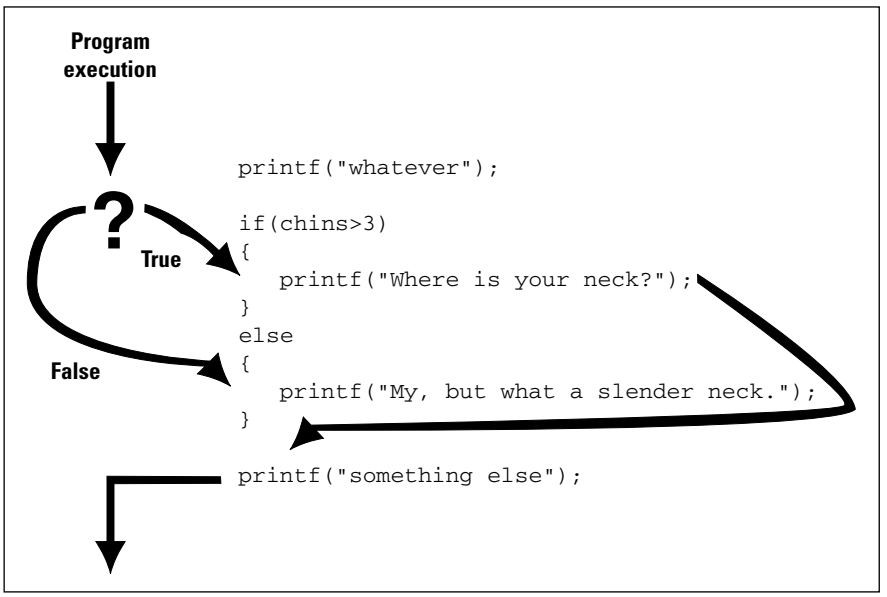

Figure 12-2: How `if` and `else` affect a program.