

Run the final result and see how much more clairvoyant the computer genie has become. Type **3** and see That number is less than 5! Type **9** and see That number is more than 5! Type **5** and the genie knows: You typed in 5!

- ✓ The `else-if` comparison resembles combined `else` and `if` statements. The second `if` comes right after `else` and a space. Then, it has its own comparison statement, which is judged either true or false.
- ✓ In `GENIE2.C`, the `else-if` comparison is `number==5`, testing to see whether the value of the `number` variable is 5. Two equal signs are used for this comparison.
- ✓ You can do `else-if`, `else-if`, `else-if` all day long, if you want. However, the C language has a better solution in the `select-case` structure. I cover this topic in this book's companion volume, *C All-in-One Desk Reference For Dummies* (Wiley).

Bonus program! The really, really smart genie

A solution always exists. If you wanted to, you could write a program that would `if`-compare any value, from zero to infinity and back again, and the “computer genie” would accurately guess it. But, why bother with `if` at all?

Okay, the `if` keyword is the subject of this section, along with `if-else` and `else-if` and so on. But, the following source code for `GENIE3.C` doesn't use `if` at all. It cheats so that the genie always guesses correctly:

```
#include <stdio.h>

int main()
{
    char num;

    printf("I am your computer genie!\n");

    printf("Enter a number from 0 to 9:");
    num = getchar();

    printf("You typed in %c!\n",num);

    printf("The genie knows all, sees all!\n");
    return(0);
}
```