The first call of `printf` writes `To C, or not to C: `. The second call writes `that is the question.` and advances to the next line. The net effect is the same as the original `printf`—the user can't tell the difference.

The new-line character can appear more than once in a string literal. To display the message

```
Brevity is the soul of wit.
  --Shakespeare
```

we could write

```
printf("Brevity is the soul of wit.\n  --Shakespeare\n");
```

2.3 Comments

Our `pun.c` program still lacks something important: documentation. Every program should contain identifying information: the program name, the date written, the author, the purpose of the program, and so forth. In C, this information is placed in *comments*. The symbol `/*` marks the beginning of a comment and the symbol `*/` marks the end:

```
/* This is a comment */
```

Comments may appear almost anywhere in a program, either on separate lines or on the same lines as other program text. Here's what `pun.c` might look like with comments added at the beginning:

```
/* Name: pun.c                    */
/* Purpose: Prints a bad pun.     */
/* Author: K. N. King             */

#include <stdio.h>

int main(void)
{
  printf("To C, or not to C: that is the question.\n");
  return 0;
}
```

Comments may extend over more than one line; once it has seen the `/*` symbol, the compiler reads (and ignores) whatever follows until it encounters the `*/` symbol. If we like, we can combine a series of short comments into one long comment:

```
/* Name: pun.c
   Purpose: Prints a bad pun.
   Author: K. N. King */
```

A comment like this can be hard to read, though, because it's not easy to see where