W *6. Write a call of `printf` that prints

`1 widget`

if the `widget` variable (of type `int`) has the value 1, and

n `widgets`

otherwise, where *n* is the value of `widget`. You are not allowed to use the `if` statement or any other statement; the answer must be a single call of `printf`.

*7. Suppose that we call `scanf` as follows:

```
n = scanf("%d%f%d", &i, &x, &j);
```

(`i`, `j`, and `n` are `int` variables and `x` is a `float` variable.) Assuming that the input stream contains the characters shown, give the values of `i`, `j`, `n`, and `x` after the call. In addition, indicate which characters were consumed by the call.

(a) `10•20•30¤`
(b) `1.0•2.0•3.0¤`
(c) `0.1•0.2•0.3¤`
(d) `.1•.2•.3¤`

W 8. In previous chapters, we've used the `scanf` format string `" %c"` when we wanted to skip white-space characters and read a nonblank character. Some programmers use `"%1s"` instead. Are the two techniques equivalent? If not, what are the differences?

Section 22.4

9. Which one of the following calls is *not* a valid way of reading one character from the standard input stream?

(a) `getch()`
(b) `getchar()`
(c) `getc(stdin)`
(d) `fgetc(stdin)`

W 10. The `fcopy.c` program has one minor flaw: it doesn't check for errors as it's writing to the destination file. Errors during writing are rare, but do occasionally occur (the disk might become full, for example). Show how to add the missing error check to the program, assuming that we want it to display a message and terminate immediately if an error occurs.

11. The following loop appears in the `fcopy.c` program:

```
while ((ch = getc(source_fp)) != EOF)
  putc(ch, dest_fp);
```

Suppose that we neglected to put parentheses around `ch = getc(source_fp)`:

```
while (ch = getc(source_fp) != EOF)
  putc(ch, dest_fp);
```

Would the program compile without an error? If so, what would the program do when it's run?

12. Find the error in the following function and show how to fix it.

```
int count_periods(const char *filename)
{
  FILE *fp;
  int n = 0;
```