Q: Do I have to use the names `argc` and `argv` for `main`'s parameters? [p. 302]

A: No. Using the names `argc` and `argv` is merely a convention, not a language requirement.

Q: I've seen `argv` declared as `**argv` instead of `*argv[]`. Is this legal?

A: Certainly. When declaring a parameter, writing `*a` is always the same as writing `a[]`, regardless of the type of `a`'s elements.

Q: We've seen how to set up an array whose elements are pointers to string literals. Are there any other applications for arrays of pointers?

A: Yes. Although we've focused on arrays of pointers to character strings, that's not the only application of arrays of pointers. We could just as easily have an array whose elements point to any type of data, whether in array form or not. Arrays of pointers are particularly useful in conjunction with dynamic storage allocation.

dynamic storage allocation ➤ *17.1*

Exercises

Section 13.3

1. The following function calls supposedly write a single new-line character, but some are incorrect. Identify which calls don't work and explain why.

```
(a) printf("%c", '\n');      (g) putchar('\n');
(b) printf("%c", "\n");      (h) putchar("\n");
(c) printf("%s", '\n');      (i) puts('\n');
(d) printf("%s", "\n");      (j) puts("\n");
(e) printf('\n');            (k) puts("");
(f) printf("\n");
```

W 2. Suppose that `p` has been declared as follows:

```
char *p = "abc";
```

Which of the following function calls are legal? Show the output produced by each legal call, and explain why the others are illegal.

```
(a) putchar(p);
(b) putchar(*p);
(c) puts(p);
(d) puts(*p);
```

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

```
scanf("%d%s%d", &i, s, &j);
```

If the user enters `12abc34 56def78`, what will be the values of `i`, `s`, and `j` after the call? (Assume that `i` and `j` are `int` variables and `s` is an array of characters.)

W 4. Modify the `read_line` function in each of the following ways:

(a) Have it skip white space before beginning to store input characters.

(b) Have it stop reading at the first white-space character. *Hint:* To determine whether or not a character is white space, call the `isspace` function.

`isspace` function ➤ *23.5*