

```
memcpy(a, b, sizeof(a));
```

Many programmers prefer `memcpy`, especially for large arrays, because it's potentially faster than an ordinary loop.

**\*Q:** Section 6.4 mentioned that C99 doesn't allow a `goto` statement to bypass the declaration of a variable-length array. What's the reason for this restriction?

**A:** The memory used to store a variable-length array is usually allocated when the declaration of the array is reached during program execution. Bypassing the declaration using a `goto` statement could result in a program accessing the elements of an array that was never allocated.

## Exercises

### Section 8.1

1. We discussed using the expression `sizeof(a) / sizeof(a[0])` to calculate the number of elements in an array. The expression `sizeof(a) / sizeof(t)`, where `t` is the type of `a`'s elements, would also work, but it's considered an inferior technique. Why?
2. The Q&A section shows how to use a *letter* as an array subscript. Describe how to use a *digit* (in character form) as a subscript.
3. Write a declaration of an array named `weekend` containing seven `bool` values. Include an initializer that makes the first and last values `true`; all other values should be `false`.
4. (C99) Repeat Exercise 3, but this time use a designated initializer. Make the initializer as short as possible.
5. The Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13, ..., where each number is the sum of the two preceding numbers. Write a program fragment that declares an array named `fib_numbers` of length 40 and fills the array with the first 40 Fibonacci numbers. *Hint:* Fill in the first two numbers individually, then use a loop to compute the remaining numbers.

### Section 8.2

6. Calculators, watches, and other electronic devices often rely on seven-segment displays for numerical output. To form a digit, such devices "turn on" some of the seven segments while leaving others "off":



Suppose that we want to set up an array that remembers which segments should be "on" for each digit. Let's number the segments as follows:

```

  0
5| 6 |1
  4| 3 |2

```

Here's what the array might look like, with each row representing one digit:

```
const int segments[10][7] = {{1, 1, 1, 1, 1, 1, 0}, ...};
```

I've given you the first row of the initializer; fill in the rest.

7. Using the shortcuts described in Section 8.2, shrink the initializer for the `segments` array (Exercise 6) as much as you can.
8. Write a declaration for a two-dimensional array named `temperature_readings` that stores one month of hourly temperature readings. (For simplicity, assume that a month has 30 days.) The rows of the array should represent days of the month; the columns should represent hours of the day.
9. Using the array of Exercise 8, write a program fragment that computes the average temperature for a month (averaged over all days of the month and all hours of the day).
10. Write a declaration for an  $8 \times 8$  char array named `chess_board`. Include an initializer that puts the following data into the array (one character per array element):

```

r n b q k b n r
p p p p p p p p
. . . . .
. . . . .
. . . . .
P P P P P P P P
R N B Q K B N R

```

11. Write a program fragment that declares an  $8 \times 8$  char array named `checker_board` and then uses a loop to store the following data into the array (one character per array element):

```

B R B R B R B R
R B R B R B R B
B R B R B R B R
R B R B R B R B
B R B R B R B R
R B R B R B R B
B R B R B R B R
R B R B R B R B

```

*Hint:* The element in row  $i$ , column  $j$ , should be the letter B if  $i + j$  is an even number.

## Programming Projects

- Modify the `repdigit.c` program of Section 8.1 so that it shows which digits (if any) were repeated:  

```

Enter a number: 939577
Repeated digit(s): 7 9

```
- Modify the `repdigit.c` program of Section 8.1 so that it prints a table showing how many times each digit appears in the number:  

```

Enter a number: 41271092
Digit:          0  1  2  3  4  5  6  7  8  9
Occurrences:    1  2  2  0  1  0  0  1  0  1

```
- Modify the `repdigit.c` program of Section 8.1 so that the user can enter more than one number to be tested for repeated digits. The program should terminate when the user enters a number that's less than or equal to 0.