

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

1. 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
```
2. 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
```
3. 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.