Programming Exercises

1. Write a program that requests the user to enter two integers. The program should then calculate and report the sum of all the integers between and including the two integers. At this point, assume that the smaller integer is entered first. For example, if the user enters 2 and 9, the program should report that the sum of all the integers from 2 through 9 is 44.

2. Redo Listing 5.4 using a type `array` object instead of a built-in array and type `long double` instead of `long long`. Find the value of 100!

3. Write a program that asks the user to type in numbers. After each entry, the program should report the cumulative sum of the entries to date. The program should terminate when the user enters 0.

4. Daphne invests $100 at 10% simple interest. That is, every year, the investment earns 10% of the original investment, or $10 each and every year:

   interest = 0.10 × original balance

   At the same time, Cleo invests $100 at 5% compound interest. That is, interest is 5% of the current balance, including previous additions of interest:

   interest = 0.05 × current balance

   Cleo earns 5% of $100 the first year, giving her $105. The next year she earns 5% of $105, or $5.25, and so on. Write a program that finds how many years it takes for the value of Cleo's investment to exceed the value of Daphne's investment and then displays the value of both investments at that time.

5. You sell the book *C++ for Fools*. Write a program that has you enter a year's worth of monthly sales (in terms of number of books, not of money). The program should use a loop to prompt you by month, using an array of `char *` (or an array of `string` objects, if you prefer) initialized to the month strings and storing the input data in an array of `int`. Then, the program should find the sum of the array contents and report the total sales for the year.

6. Do Programming Exercise 5 but use a two-dimensional array to store input for 3 years of monthly sales. Report the total sales for each individual year and for the combined years.

7. Design a structure called `car` that holds the following information about an automobile: its make, as a string in a character array or in a `string` object, and the year it was built, as an integer. Write a program that asks the user how many cars to catalog. The program should then use `new` to create a dynamic array of that many `car` structures. Next, it should prompt the user to input the make (which might consist of more than one word) and year information for each structure. Note that this requires some care because it alternates reading strings with numeric data (see

Chapter 4). Finally, it should display the contents of each structure. A sample run should look something like the following:

```
How many cars do you wish to catalog? 2
Car #1:
Please enter the make: Hudson Hornet
Please enter the year made: 1952
Car #2:
Please enter the make: Kaiser
Please enter the year made: 1951
Here is your collection:
1952 Hudson Hornet
1951 Kaiser
```

8. Write a program that uses an array of `char` and a loop to read one word at a time until the word `done` is entered. The program should then report the number of words entered (not counting `done`). A sample run could look like this:

```
Enter words (to stop, type the word done):
anteater birthday category dumpster
envy finagle geometry done for sure
You entered a total of 7 words.
```

You should include the `cstring` header file and use the `strcmp()` function to make the comparison test.

9. Write a program that matches the description of the program in Programming Exercise 8, but use a `string` class object instead of an array. Include the `string` header file and use a relational operator to make the comparison test.

10. Write a program using nested loops that asks the user to enter a value for the number of rows to display. It should then display that many rows of asterisks, with one asterisk in the first row, two in the second row, and so on. For each row, the asterisks are preceded by the number of periods needed to make all the rows display a total number of characters equal to the number of rows. A sample run would look like this:

```
Enter number of rows: 5
....*
...**
..***
.****
*****
```