

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 3.4 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:

$$\text{interest} = 0.10 \times \text{original balance}$$

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:

$$\text{interest} = 0.05 \times \text{current balance}$$

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 2). 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
```

```
....*
...**
..***
.****
*****
```

11. Write a program that reads keyboard input to the @ symbol and that echoes the input except for digits, converting each uppercase character to lowercase, and vice versa. (Don't forget the `cctype` family.)
12. Write a program that reads up to 10 donation values into an array of `double`. (Or, if you prefer, use an array template object.) The program should terminate input on non-numeric input. It should report the average of the numbers and also report how many numbers in the array are larger than the average.
13. Write a precursor to a menu-driven program. The program should display a menu offering four choices, each labeled with a letter. If the user responds with a letter other than one of the four valid choices, the program should prompt the user to enter a valid response until the user complies. Then the program should use a switch to select a simple action based on the user's selection. A program run could look something like this:

```
Please enter one of the following choices:
```

```
c) carnivore          p) pianist
t) tree               g) game
f
```

```
Please enter a c, p, t, or g: q
```

```
Please enter a c, p, t, or g: t
```

```
A maple is a tree.
```

14. When you join the Benevolent Order of Programmers, you can be known at BOP meetings by your real name, your job title, or your secret BOP name. Write a program that can list members by real name, by job title, by secret name, or by a member's preference. Base the program on the following structure:

```
// Benevolent Order of Programmers name structure
```

```
struct bop {
    char fullname[STRSIZE]; // real name
    char title[STRSIZE];    // job title
    char bopname[STRSIZE];  // secret BOP name
    int preference;         // 0 = fullname, 1 = title, 2 = bopname
};
```

In the program, create a small array of such structures and initialize it to suitable values. Have the program run a loop that lets the user select from different alternatives:

- a. display by name b. display by title
- c. display by bopname d. display by preference
- q. quit

Note that “display by preference” does not mean display the preference member; it means display the member corresponding to the preference number. For instance, if preference is 1, choice d would display the programmer’s job title. A sample run may look something like the following:

```
Benevolent Order of Programmers Report
a. display by name      b. display by title
c. display by bopname   d. display by preference
q. quit
Enter your choice: a
Wimp Macho
Raki Rhodes
Celia Laiter
Hoppy Hipman
Pat Hand
Next choice: d
Wimp Macho
Junior Programmer
MIPS
Analyst Trainee
LOOPY
Next choice: q
Bye!
```

15. The Kingdom of Neutronia, where the unit of currency is the tvarp, has the following income tax code:

First 5,000 tvarps: 0% tax

Next 10,000 tvarps: 10% tax

Next 20,000 tvarps: 15% tax

Tvarps after 35,000: 20% tax

For example, someone earning 38,000 tvarps would owe $5,000 \times 0.00 + 10,000 \times 0.10 + 20,000 \times 0.15 + 3,000 \times 0.20$, or 4,600 tvarps. Write a program that uses a loop to solicit incomes and to report tax owed. The loop should terminate when the user enters a negative number or non-numeric input.

16. Put together a program that keeps track of monetary contributions to the Society for the Preservation of Rightful Influence. It should ask the user to enter the number

of contributors and then solicit the user to enter the name and contribution of each contributor. The information should be stored in a dynamically allocated array of structures. Each structure should have two members: a character array (or else a `string` object) to store the name and a `double` member to hold the amount of the contribution. After reading all the data, the program should display the names and amounts donated for all donors who contributed \$10,000 or more. This list should be headed by the label `Grand Patrons`. After that, the program should list the remaining donors. That list should be headed `Patrons`. If there are no donors in one of the categories, the program should print the word “none.” Aside from displaying two categories, the program need do no sorting.

17. Write a program that reads input a word at a time until a lone `q` is entered. The program should then report the number of words that began with vowels, the number that began with consonants, and the number that fit neither of those categories. One approach is to use `isalpha()` to discriminate between words beginning with letters and those that don't and then use an `if` or `switch` statement to further identify those passing the `isalpha()` test that begin with vowels. A sample run might look like this:

```
Enter words (q to quit):
```

```
The 12 awesome oxen ambled  
quietly across 15 meters of lawn. q
```

```
5 words beginning with vowels
```

```
4 words beginning with consonants
```

```
2 others
```

18. Write a program that opens a text file, reads it character-by-character to the end of the file, and reports the number of characters in the file.
19. Do Programming Exercise 6 but modify it to get information from a file. The first item in the file should be the number of contributors, and the rest of the file should consist of pairs of lines, with the first line of each pair being a contributor's name and the second line being a contribution. That is, the file should look like this:

```
4
```

```
Sam Stone
```

```
2000
```

```
Freida Flass
```

```
100500
```

```
Tammy Tubbs
```

```
5000
```

```
Rich Raptor
```

```
55000
```