Programming Exercises

1. 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.)

2. 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.

3. 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.
```

4. 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!
```

5. 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.

6. 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.

7. 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
```

8. 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.

9. 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
```