the return value to a `char` variable or to a `char*` variable.

.6: Write a function template that returns the larger of its two arguments.

.7: Given the template of Review Question 6 and the box structure of Review Question 4, provide a template specialization that takes two box arguments and returns the one with the larger volume.

Programming Exercises

1: Write a function that normally takes one argument, the address of a string, and prints that string once. However, if a second, type int argument is provided and is nonzero, the function prints the string a number of times equal to the number of times that function has been called to at that point. (Note that the number of times the string is printed is not equal to the value of the second argument; it's equal to the number of times the function has been called.) Yes, this is a silly function, but it makes you use some of the techniques discussed in this chapter. Use the function in a simple program that demonstrates how the function works.

2: The `CandyBar` structure contains three members. The first member holds the brand name of a candy bar. The second member holds the weight (which may have a fractional part) of the candy bar, and the third member holds the number of calories (an integer value) in the candy bar. Write a program that uses a function that takes as arguments a reference to a `CandyBar`, a pointer-to-char, a `double`, and an `int` and uses the last three values to set the corresponding members of the structure. The last three arguments should have default values of "Millennium Munch," 2.85, and 350. Also, the program should use a function taking a reference to a `CandyBar` as an argument and display the contents of the structure. Use `const` where appropriate.

3: Following is a program skeleton. Complete it by providing the described functions and prototypes. Note that there should be two `show()` functions, each using default arguments. Use `const` arguments when appropriate. Note that