Program Notes

To create a string of \( n \) visible characters, you need storage for \( n + 1 \) characters in order to have space for the null character. So the function asks for \( n + 1 \) bytes to hold the string. Next, it sets the final byte to the null character. Then, it fills in the rest of the array from back to front. The loop

\[
\text{while (n-- > 0)}
\]
\[\text{pstr}[n] = c;\]

cycles \( n \) times as \( n \) decreases to zero, filling \( n \) elements. At the start of the final cycle, \( n \) has the value 1. Because \( n-- \) means use the value and then decrement it, the while loop test condition compares 1 to 0, finds the test to be true, and continues. But after making the test, the function decrements \( n \) to 0, so \( \text{pstr}[0] \) is the last element set to \( c \). The reason for filling the string from back to front instead of front to back is to avoid using an additional variable. Using the other order would involve something like this:

\[
\text{int i = 0;}
\]
\[
\text{while (i < n)}
\]
\[\text{pstr}[i++] = c;\]

Note that the variable \( \text{pstr} \) is local to the \texttt{buildstr} function, so when that function terminates, the memory used for \( \text{pstr} \) (but not for the string) is freed. But because the function returns the value of \( \text{pstr} \), the program is able to access the new string through the \( \text{ps} \) pointer in \texttt{main}().

The program uses \texttt{delete} to free memory used for the string after the string is no longer needed. Then, it reuses \( \text{ps} \) to point to the new block of memory obtained for the next string and frees that memory. The disadvantage to this kind of design (having a function return a pointer to memory allocated by \texttt{new}) is that it makes it the programmer’s responsibility to remember to use \texttt{delete}. The \texttt{auto_ptr} template, discussed in Chapter 16, can help automate the process.

Functions and Structures

Let's move from arrays to structures. It's easier to write functions for structures than for