```cpp
continue;
cout << "Bad input; input process terminated.\n";
break;
}
else if (temp < 0)     // signal to terminate
    break;
ar[i] = temp;
}
return i;
}

Note that the code includes a prompt to the user in the program. If the user enters a non-negative value, the value is assigned to the array. Otherwise, the loop terminates. If the user enters only valid values, the loop terminates after it reads limit values. The last thing the loop does is increment i, so after the loop terminates, i is 1 greater than the last array index, hence it's equal to the number of filled elements. The function then returns that value.

** Showing the Array and Protecting It with const **

Building a function to display the array contents is simple. You pass the name of the array and the number of filled elements to the function, which then uses a loop to display each element. But there is another consideration—guaranteeing that the display function doesn't alter the original array. Unless the purpose of a function is to alter data passed to it, you should safeguard it from doing so. That protection comes automatically with ordinary arguments, because C++ passes them by value and the function works with a copy. But functions that use an array work with the original. After all, that's why the fill_array() function is able to do its job. To keep a function from accidentally altering the contents of an array argument, you can use the keyword const (discussed in Chapter 3, "Dealing with Data") when you declare the formal argument:

```cpp
void show_array(const double ar[], int n);
```

The declaration states that the pointer ar points to constant data. This means that you can't use ar to change the data. That is, you can use a value such as ar[0], but you can't change that value. Note that this doesn't mean that the original array need be constant; it just means that you can't use ar in the show_array() function to change the data. Thus,

This document was created by an unregistered ChmMagic, please go to http://www.bisenter.com to register it. Thanks