Is the correspondence between array names and pointers a good thing? Indeed, it is. The design decision to use array addresses as arguments saves the time and memory needed to copy an entire array. The overhead for using copies can be prohibitive if you're working with large arrays. Not only does a program need more computer memory, but it has to spend time copying large blocks of data. On the other hand, working with the original data raises the possibility of inadvertent data corruption. That's a real problem in classic C, but ANSI C and C++'s const modifier provides a remedy. We'll soon show an example. But first, let's alter Listing 7.5 to illustrate some points about how array functions operate. Listing 7.6 demonstrates that cookies and arr have the same value. It also shows how the pointer concept makes the sum_arr function more versatile than it may have appeared at first.

Listing 7.6 arrfun2.cpp

// arrfun2.cpp -- functions with an array argument