Then the two pointers `elboud` and `elboud + 20` define the range. First, `elboub`, being the name of the array, points to the first element. The expression `elboud + 19` points to the last element (that is, `elboud[19]`), so `elboud + 20` points to one past the end of the array. Passing a range to a function tells it which elements to process. Listing 7.8 modifies Listing 7.6 to use two pointers to specify a range.

Listing 7.8 arrfun4.cpp

```
// arrfun4.cpp -- functions with an array range
#include <iostream>
using namespace std;
const int ArSize = 8;
int sum_arr(const int * begin, const int * end);
int main()
{
    int cookies[ArSize] = {1,2,4,8,16,32,64,128};
    // some systems require preceding int with static to
    // enable array initialization

    int sum = sum_arr(cookies, cookies + ArSize);
    cout << "Total cookies eaten: " << sum << "\n";
    sum = sum_arr(cookies, cookies + 3);       // first 3 elements
    cout << "First three eaters ate " << sum << " cookies.\n";
    sum = sum_arr(cookies + 4, cookies + 8);    // last 4 elements
    cout << "Last four eaters ate " << sum << " cookies.\n";
    return 0;
}

// return the sum of an integer array
int sum_arr(const int * begin, const int * end)
{
    const int * pt;
    int total = 0;
    for (pt = begin; pt != end; pt++)
        total = total + *pt;
    return total;
```