APPENDIX

L Passing Command Line Arguments

If you are working in a command line environment such as UNIX, Linux, or the DOS prompt, it might be helpful to write programs that take arguments from the command line. For example, suppose we have a program called sum, which takes two numbers as command line arguments and displays their sum. We could enter the following command at the operating system prompt:

```
sum 12 16
```

The arguments, which are separated by a space, are 12 and 16. Because a C++ program starts its execution with function main, command line arguments are passed to main. Function main can be optionally written with two special parameters, which are traditionally named argc and argv. The argc parameter is an int, and the argv parameter is an array of char pointers. Here is an example function header for main, using these two parameters:

```
int main(int argc, char *argv[])
```

The argc parameter contains the number of items that were typed on the command line, including the name of the program. For example, if the sum program described above is executed with the command sum 12 16, the argc parameter will contain 3.

As previously mentioned, the argv parameter is an array of char pointers. In the function header, the brackets are empty because argv is an external array of unknown size. The number that is stored in argc, however, will be the number of elements in the argv array. Each pointer in the argv array points to a C-string holding a command line argument. Once again, assume the sum program is executed with the command sum 12 16. The elements of the argv array will reference the items on the command line in the following manner:

```
argv[0] = "sum"
argv[1] = "12"
argv[2] = "16"
```

Before we look at the code for the sum program, let's look at Program L-1. It is a short program that simply displays its command line arguments. (The program is named argdemo.cpp.)

Program L-1 (argdemo.cpp)

```
// This program demonstrates how to read
// command line arguments.
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
   cout << "You entered " << (argc - 1)
   cout << " command line arguments.\n";
   if (argc > 1)
   {
      cout << "Here they are:\n";
      for (int count = 1; count < argc; count++)
         cout << argv[count] << endl;
   }
   return 0;
}
```

Example Session on a UNIX System with Example Input Shown In Bold

```
$ argdemo Hello World [Enter]
You entered 2 command line arguments.
Here they are:
Hello
World
$
```

Now, let's look at the code for the sum program.

Program L-2 (sum.cpp)

```
// This program takes two command line arguments,
// assumed to be numbers, and displays their sum.
#include <iostream>
#include <cmath>      // Needed for atof
using namespace std;

int main(int argc, char *argv[])
{
   double total = 0;

   if (argc > 1)
   {
      for (int count = 1; count < argc; count++)
         total += atof(argv[count]);
      cout << total << endl;
   }
   return 0;
}
```

Example Session on a UNIX System with Example Input Shown In Bold

```
$ sum 12 16[Enter]
28
$ sum 1 2 3 4 5[Enter]
15
$
```