```cpp
//---------------------------------------------------------------------------
// MPI_Bcast
//---------------------------------------------------------------------------
#if (defined __linux__) || (defined _AIX)
   #include <sys/types.h>
   #include <sys/stat.h>
   #include <unistd.h>
#elif (defined _WIN32) || (defined _WIN64)
   #include <conio.h>
   #include <direct.h>
#endif

#include <mpi.h>
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
using namespace std;
//---------------------------------------------------------------------------
int main(int argc, char* argv[])
{
   int i, rank, size, mess1 = 0, mess2[100];
   double time;

   MPI_Init(&argc, &argv);
   MPI_Barrier(MPI_COMM_WORLD);
   time = -MPI_Wtime();

   MPI_Comm_size(MPI_COMM_WORLD, &size);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);

   for (i = 0; i < size; i++)
      mess2[i] = 0;

   if (rank == 0) {
      printf("\nThere are %d processors.\n", size);
      mess1 = 333;
      std::fflush(stdout);
   }

   if (rank == 0) {
      cout << "\nInitial values of array mess2 are: ";
      for (i = 0; i < size; i++)
         cout << mess2[i] << " ";
      cout << endl;
   }
   fflush(stdin);

   MPI_Bcast(&mess1, 1, MPI_INT, 0, MPI_COMM_WORLD);

   MPI_Gather(&mess1, 1, MPI_INT, mess2, 1, MPI_INT, 0, MPI_COMM_WORLD);

   if (rank == 0) {
      cout << "\nFinal values of array mess2 are: ";
```

```
54        for (i = 0; i < size; i++)
55           cout << mess2[i] << " ";
56        cout << endl;
57     }
58
59     time += MPI_Wtime();
60     if (rank == 0)
61        printf("\nTime = %10.6f s\n", time);
62
63     MPI_Finalize();
64     return 0;
65  }
```