

```
1:
2: #include <iostream>                                // MPI
3: #include <mpi.h>
4: #include <vector>
5: using namespace std;
6:
7:
8: void DebugVector(vector<double> const& xin, MPI_Comm const& icomm)
9: {
10:     int myrank, numprocs;
11:
12:     MPI_Comm_rank(icmp, &myrank);
13:     MPI_Comm_size(icmp, &numprocs);
14:     int selectedProcess = 0;
15:     while(selectedProcess >= 0 && selectedProcess < numprocs)
16:     {
17:         if(myrank == 0)
18:         {
19:             cout << "Enter process number: " << endl;
20:             cin >> selectedProcess;
21:         }
22:
23:         MPI_Bcast(&selectedProcess, 1, MPI_INT, 0, icomm); ✓
24:         MPI_Barrier(icmp);
25:         if(myrank == selectedProcess ) ✓
26:         {
27:             for(double x : xin)
28:             {
29:                 cout << x << " ";
30:             }
31:             cout << endl;
32:         }
33:         MPI_Barrier(icmp); ✓
34:     }
35: }
36:
37: }
38:
39: double par_scalar(const vector<double>& x, const vector<double>& y, MPI_Comm icomm)
40: {
41:     double local = 0.0;
42:     for (unsigned int i = 0; i < x.size(); ++i) {
43:         local += x[i] * y[i];
44:     }
45:
46:     double sum = 0.0; ✓
47:     MPI_Allreduce(&local, &sum, 1, MPI_DOUBLE, MPI_SUM, icomm);
48:
49:     return sum;
50: }
51:
52: void E7(const vector<double>& x, MPI_Comm comm,
53:         double& min, double& max)
54: {
55:     double local_min = x[0];
56:     double local_max = x[0];
57:
58:     for (double x_i : x) {
59:         if (x_i < local_min)
60:         {
61:             local_min = x_i;
62:         }
63:         if (x_i > local_max)
64:         {
65:             local_max = x_i;
66:         }
67:     }
```

```
68:     }
69:
70:     MPI_Allreduce(&local_min, &min, 1, MPI_DOUBLE, MPI_MIN, comm);
71:     MPI_Allreduce(&local_max, &max, 1, MPI_DOUBLE, MPI_MAX, comm);
72: }
73:
74:
75: int main(int argc, char *argv[])
76: {
77:     MPI_Comm icomm = MPI_COMM_WORLD;
78:     MPI_Init(&argc, &argv);
79:     int myrank, numprocs;
80:     //numprocs = 1; // delete this line when uncommenting the next line
81:     MPI_Comm_rank(icom, &myrank); // my MPI-rank
82:     MPI_Comm_size(icom, &numprocs);
83:
84:     if (0==myrank) {
85:         cout << "\n There are " << numprocs << " processes running.\n \n";
86:     }
87:
88:
89:     if (0==myrank) cout << endl;
90:
91:     unsigned int n=20;
92:     vector<double> x(n);
93:     vector<double> y(n);
94:     double x_i;
95:
96:     for(unsigned int i=0; i<n; i++)
97:     {
98:         x_i = myrank*100+(i%5)*10+i; // E8
99:         x[i] = x_i;
100:
101:         if(myrank == 0 && i == 0)
102:         {
103:             y[i] = 0.0;
104:         }
105:         else{
106:             y[i] = 1.0/x_i;
107:         }
108:
109:     }
110:
111:     DebugVector(x,icom);
112:
113:     double result = par_scalar(x,y,icom);
114:
115:     if(myrank == 0)
116:     {
117:         cout << "result scalar: " << result << endl;
118:     }
119:
120:     double min, max;
121:     E7(x,icom,min,max);
122:     if(myrank == 0)
123:     {
124:         cout << "min: " << min << ", max: " << max << endl;
125:     }
126:
127:     E8
128:
129:     MPI_Alltoall(MPI_IN_PLACE, 0, MPI_DOUBLE, x.data(), 5, MPI_DOUBLE, icom);
130:     DebugVector(x,icom);
131:
132:     MPI_Finalize();
133:     return 0;
134: }
135:
```

Exchange min/max!?

136: