

```
1 //-----
2 // 2C_Sum_of_Numbers.cpp
3 //-----
4 #if (defined __linux__) || (defined _AIX)
5 #include <sys/types.h>
6 #include <sys/stat.h>
7 #include <unistd.h>
8 #elif (defined _WIN32) || (defined _WIN64)
9 #include <conio.h>
10 #include <direct.h>
11 #endif
12
13 #include <mpi.h>
14 #include <iostream>
15 using namespace std;
16
17 #define N 10000
18
19 int main(int argc, char** argv) {
20     int i, rank, size, start_number, end_number, borders[64][2];
21     long unsigned sum = 0, total_sum = 0, accum[64];
22     double time;
23     MPI_Status status;
24
25     MPI_Init(&argc, &argv);
26     MPI_Barrier(MPI_COMM_WORLD);
27     time = -MPI_Wtime();
28
29     MPI_Comm_size(MPI_COMM_WORLD, &size);
30     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
31     if (rank == 0)
32         printf("\nThere are %d processes.\n", size);
33
34     start_number = rank * N / size + 1;
35     end_number = (rank + 1) * N / size;
36
37     for (i = start_number; i <= end_number; i++)
38         sum += i;
39
40     if (rank != 0) {
41         MPI_Send(&start_number, 1, MPI_INT, 0, 0, MPI_COMM_WORLD);
42         MPI_Send(&end_number, 1, MPI_INT, 0, 1, MPI_COMM_WORLD);
43         MPI_Send(&sum, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
44     }
45     else {
46         borders[0][0] = start_number;
47         borders[0][1] = end_number;
48         total_sum = accum[0] = sum;
49         for (i = 1; i < size; i++) {
50             MPI_Recv(&borders[i][0], 1, MPI_INT, i, 0, MPI_COMM_WORLD, &status);
51             MPI_Recv(&borders[i][1], 1, MPI_INT, i, 1, MPI_COMM_WORLD, &status);
52             MPI_Recv(&accum[i], 1, MPI_INT, i, 2, MPI_COMM_WORLD, &status);
53             total_sum += accum[i];

```

```
54     }
55 }
56
57 if (rank == 0) {
58     for (i = 0; i < size; i++)
59         cout << "\nProcess " << i << ": The sum of numbers from " << borders[i][0]
60         << " to " << borders[i][1] << " is: " << accum[i];
61
62     cout << "\n\nThe sum of numbers from 1 to " << N << " is: " << total_sum;
63 }
64 time += MPI_Wtime();
65 if (rank == 0)
66     printf("\n\nTime = %10.6f s\n", time);
67
68 fflush(stdout);
69 MPI_Finalize();
70
71 return 0;
72 }
```