```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <mpi.h>

int main(int argc, char *argv[]) {
  if (argc != 2) {
    printf("Usage : allreduce message_size\n");
    return 1;
  }

  int rank;
  int size = atoi(argv[1]);
  char input_buffer[size];
  char recv_buffer[size];

  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  int i;
  srand(time(NULL));
  for (i = 0; i < size; i++)
    input_buffer[i] = rand() % 256;
  double total_time = 0.0;
  double start_time = 0.0;
  for (i = 0; i < 100; i++) {
    MPI_Barrier(MPI_COMM_WORLD);
    start_time = MPI_Wtime();

MPI_Allreduce(input_buffer,recv_buffer,size,MPI_BYTE,MPI_BOR,MPI_COMM_
WORLD);
    MPI_Barrier(MPI_COMM_WORLD);
    total_time += (MPI_Wtime() - start_time);
  }

  if (rank == 0) {
    printf("Average time for allreduce : %f secs\n", total_time/100);
  }

  MPI_Finalize();
}
```