# 1 Usage

make coordinator
./coordinator <N> <Root> <Verbose:Def=1> <Infile:Def='in'> <Outfile:Def='out'>
Sample use $ ./coordinator 5 3

Parameters:
N - number of integers to read and number of nodes to make.
Root - the node in the ring to which coordinator is connected.
Verbose - if 1, print messages.
Infile - read N space separated integers from here.
Outfile - save sorted output here.

# 2 Implementation

Every "message" sent follows the format:
<Dest_id> <Source_id> <Size> <N1> <N2> ... <NSize>
Coordinator id is 0, others are successive numbers from root until N and then from 1 to N-1.

Upon completeion, coordinator sends a message with Destination 0. This is treated as the terminate command. Upon receiving this, nodes will pass it on, wait for some time, then close.

Works well upto N = 256. Components may sleep for some time, proportional to N, upto 3 seconds at N=256. The same socket may have two aliases, one for 0.0.0.0 and one for localhost.