Input→Master	Master→Server	Comments
$\langle id \rangle$ start $\langle n \rangle$ $\langle port \rangle$	—	master starts a process with `./process id n port`
exit	—	master calls `./stopall` then exits
sleep $\langle n \rangle$	—	master sleeps for n milliseconds
$\langle id \rangle$ crash	—	The master crashes the given process using a kill signal. Your `process` must be compatible with this.
$\langle id \rangle$ get	get	the receiver responds to the master with its message log
$\langle id \rangle$ alive	alive	the receiver responds to the master with the ids all processes it thinks are alive
$\langle id \rangle$ broadcast $\langle message \rangle$	broadcast $\langle message \rangle$	the receiver broadcasts the given message to everyone alive, including themselves
—	connect $\langle host \rangle$:$\langle port \rangle$	connect to host:port

Table 1: Table of commands. The left column shows commands provided as input to the master; the center column the corresponding commands issued by the master to the servers.

Server→Server	Comments
message $\langle message \rangle$	the receiver registers a message and does nothing with it