

**University of Texas at Dallas—Department of Computer Science**  
**CS 6380.001 Distributed Computing—Spring 2019**  
**Project 2 Description**

Use the simulator you developed in Project 1.

Your simulation will simulate the randomized algorithm for consensus despite links failures. The code (algorithm) executed by all processes (threads) must be the same (except for the leader).

The input for this problem consists of (1)  $n$  (the number of processes of the distributed system which is equal to the number of threads to be created), (2)  $r$  (the number of rounds the algorithm is to run for), (3)  $x$ , an integer whose significance is described below; and (4) one array  $inputval[n]$  of size  $n$ ; the  $i^{th}$  element of this array gives the input value of the process  $i$ . The master thread reads input file `input.dat` containing these four inputs and then spawns  $n$  threads. The file `input.dat` is a text file and process ids are equal to the indices. [process ids are 1, 2, ...,  $n$  and  $i$ th process had id  $i$ .]. Between every pair of processes, there is a direct link. [Communication network is a completely connected graph.]

Every process knows  $n$ , process 1 is the leader and every  $x$ th message is lost (not delivered). For message loss, all messages are numbered as follows: In the first round, the message from process 1 to process 2 is message number 1, the message from process 1 to process 3 is second message, message from process 1 to process  $n$  is the  $n-1^{st}$  message, the message from process 2 to process 1 is the  $n$ th message, etc. This numbering is only for the purpose of deciding which message will be dropped at any round. These “counters” are NOT reset at the end of each round. At the end of executing the algorithm, each process should output its id, its decision value, and its level vector. Make sure that each process’s output is printed “in tact.” (One process’s output cannot be interleaved with that of another process’s output.)

Upload one tar file containing your source code, a README file that tells us how to compile and run, the sample input file (`input.dat`) and the result of running your program (script file output) on your sample input file.

Due date: March 28, 2019, 11:55 pm.