BONUS

Imthiaz Hussain (imthiazh.hussain@ufl.edu)
Lohit Bhambri (lohit.bhambri@ufl.edu)

Problem Statement:

The objective for the project is to implement **failure** in the Chord algorithm.

1. Chord is a protocol and algorithm for a peer-to-peer distributed hash table.
2. A distributed hash table stores key-value pairs by assigning keys to different nodes; a node will store the values for all the keys for which it is responsible.
3. Chord specifies how keys are assigned to nodes, and how a node can discover the value for a given key by first locating the node responsible for that key.

Run Bonus Project:

Run the following commands for the project

1. Compile the project

```
c(node).

c(main).

c(hopCalculator).
```

2. Call the main function

```
main:chord_start(1000,3).
```

where 1000 represents NumberofNodes and 3 represents NumberOfRequests.

Explanation:

Right now the chord is up and running. Now we are implicitly killing the node after distributing the finger table to the respective nodes. Suppose we are right now at the finger table of node P and a message has to be delivered to the node Z. Since node Z is dead actor, in this scenario node P will propagate the message to the neighbour that is 1 index below (let say node W is 1 index below than node Z in the finger table) that takes responsibility to deliver the message to the best possible alternative node.

Largest Achievable Network:

The largest achievable network is of 1000 nodes with average time of 8.6 (approx) seconds.

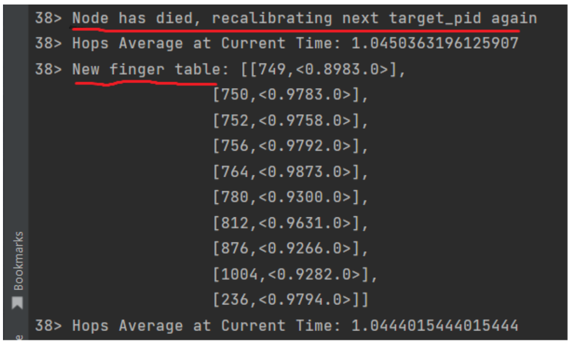
Result Screenshot:

Input:

```
main:chord_start(1000,3)
```

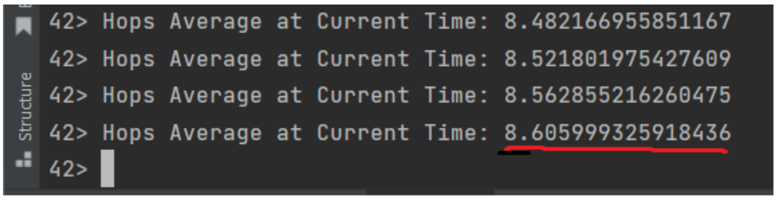
Output:

Intermediate Output during Execution: (stabilizing if node is dead)



```
38> Node has died, recalibrating next target_pid again
38> Hops Average at Current Time: 1.0450363196125907
38> New finger table: [[749,<0.8983.0>],
                       [750,<0.9783.0>],
                       [752,<0.9758.0>],
                       [756,<0.9792.0>],
                       [764,<0.9873.0>],
                       [780,<0.9300.0>],
                       [812,<0.9631.0>],
                       [876,<0.9266.0>],
                       [1004,<0.9282.0>],
                       [236,<0.9794.0>]]
38> Hops Average at Current Time: 1.0444015444015444
```

Final Result By program:
The algorithm ends in a consistent state.



```
42> Hops Average at Current Time: 8.482166955851167
42> Hops Average at Current Time: 8.521801975427609
42> Hops Average at Current Time: 8.562855216260475
42> Hops Average at Current Time: 8.605999325918436
42> 
```