Programming Assignment 2:

Prerequisite:

- The working environment should have Java-1.8 installed.
- Apache Ant-1.9.3 and above.

Steps to Execute:

- Extract PROG2_KALE_ADITYA.tar to a particular location in your Linux environment. Go to above extracted location from terminal. Go to project **DistributedHashTable**.
- Execute **command / DistributedHashTable** $ **ant**
  Build.xml gets execute and generates a jar. A **dist** folder is generated which contains **DistributedHashTable.jar.**

Steps to start Multiple Clients.

- Go to folder resources and open **config.properties** update noOfservers variable value. It is the number of Clients that will act as a server to put values in hashtable.

  Ex: **noOfServers** = 8

- Update all serverIp variable value with Ipaddress of those machine which will be used in this system.

  Ex: **serverIp1** = 192.168.239.131

  serverIp2 = 192.168.139.132

- Update all serverPort variable value with port numbers which are available for communication on above mentioned machines. Make sure that serverport1 is a port used by machine serverIp1 for communication.

  Ex: **srerverPort1** = 9991

  srerverPort2 = 8787

  Note: If you want to start more than 8 clients then you need to add serverIp and serverPort variables similar to given variables in config.properties.

Steps to run multiple Clients.

- Copy the same project structure and configuration to multiple Linux machine which are connected in a single network.
- Before starting every client you need to update the variable in config.properties **currentMachinePort** to a port number of that particular machine.
  Ex: **currentMachinePort** = 9991

- Go to above project location from terminal and execute following commands
- For Linux Environment
  - Execute **command / DistributedHashTable** $ **./runClient.sh**
    - Clients gets Start and ready to accept requests from another clients.

Note: Before starting any operations you need to check that all clients are up and ready to listen request from another clients. Otherwise system will not work.

- Now you can give input from client.
  1. **Put Element in DHT→** It will accept key and value from user and put it in a hashtable of some machine
  2. **Get Element from DHT→**. It will accept a key from user and returns its corresponding value to a user from the hashtable where it was stored while putting.
  3. **Delete Element from DHT →**It will delete that key from the hashtable where it was stored while putting.
  4. **Stop Client →** Will terminate the program.
  5. Also can Use **ctrl + C** to terminate the program.

For Windows Environment

- Execute **command \ DistributedHashTable** > **java –cp dist\ DistributedHashTable.jar edu.dht.Client**
  - Client gets start and ready to listen other peer request.