Hello,  
execute these commands one by one:  
commands are lines starting with a '>'  
  
0. if AutoCompile doesn't work as expected, execute make again  
> make clean && make  
  
1. start an AggregationServer (u can stay in current directory where makefile is)  
#for starting a default AS on port 4567, run the command below:(There is a space before the 2nd quotation mark. Don't delete it)  
> make "AggregationServer "  
  
#for starting a specific AS on a specific port:  
> make "AggregationServer <port>"  
  
\*\*WARNING before making connection:\*\*  
Don't try to connect to your server through your Public IP when you run the client and server on the same machine and the same network, which will result in a client timeout exception.  
Instead, use Private IP for local connection.  
  
2. start a ContentServer to connect some AS  
#connect to a local default AS:(There is a space before the 2nd quotation mark. Don't delete it)  
> make "ContentServer "  
  
#connect to a remote specific AS:  
> make "ContentServer <IP>:<port>"  
  
3. start a GETClient  
#connect to a local default AS:(There is a space before the 2nd quotation mark. Don't delete it)  
> make "GETClient "  
  
#connect to a remote specific AS:  
> make "GETClient <IP>:<port>"  
  
4. u can type "put" in ContentServer window, then it will read local file, generate a PUT message and send it to AggregationServer  
> put  
  
5. u can type "get" in GETClient window, then it will generate a GET message, send it to AggregationServer and parse received message from AggregationServer. The response could be a correct weather info or error code.  
> get  
#No stationID. It will get the newest entry from AS.  
> get <stationID>  
#With stationID. It will get the newest entry of the specific weather station from AS  
  
6. Now, u can type "put" or "get" in their own windows in any order u want.  
 you can see output details in server's window or clients' windows.  
  
\*\*Explanation\*\*:  
 a. Once AS received a PUT request and executed successfully, AS will send a Heartbeat Check to this CS every 15s  
 meanwhile attach its Lamport Clock with this message to CS for synchronization.  
 b. AS has a PriorityQueue for save the weather feed temporarily. Everytime it add a new entry it will save the info +  
 Lamport clock.  
 c. AS also has a cache.txt for storing these entries same as the PriorityQueue in case of crash.  
 d. Clients will always get the weather entry with the maximum Lamport Clock number, which is Long type.  
 e. If PriorityQueue's size>20, it will move the older entries into a backup.txt and only keep the 20 newest ones in  
 the queue.  
 f. if a CS give no response to two heartbeat checks then all of its sent entries will be cleared.