**Pseudocode for Project 3**

**Initialization**

Include stdio.h stdlib.h pthread.h

Define NUMBER OF WORKER THREADS

Stubs for workerThread, LogThread, DictInIt, initVars, addtosockq, remfromsockq, addtoconnq, remfromconnq, checkwordinDic(2.5), threadspawn (see loops and ECs below slide 1.9), bufferput, buffer get

Parse input of user with server and port

**InitDictionary**

If user changes the source test

Open that file and set to the var dictionary

Else set file to default

**initVariables**

Use dictionary to set int max word length so we can prevent buffer overflow later

Set port to user input or default

initalize log and socket buffer queues (slide 1.14) with respective useptr, fillptr, and slotcount

**network initialization**

**main**

initialize loop counter for thread creation

initialize array of thread IDs

while(1)

get connection file descriptor

error check //slide 2.4

add socket to ConnectionQueue //slide 1.19

try\_lock so only main can access the queue

check condition of full buffer

if yes🡪 wait on ‘empty’ condition //blocks this thread until workers wake by removing from the socketbuffer

pthread\_cond\_signal(&fillSock, &ConnectionLock) //report theres a connection in the queue to the worker threads

pthread\_mutex\_unlock( &ConnectionLock) //Release lock between main and workers

call loop to start wthreads

error check

start log thread

error check

loop to join work threads //before the server finishes an iteration, make sure threads have finished reading or writing

error check

join log threads

error check