COP 6611
Reader Writer Program

Due: March 8 2022 at 9:00a.m.

Implement a solution via semaphores and threads to the n reader 1 writer problem. Fairness always matters. You will accept the number of readers from the command line. In no case will more than 18 readers be used and always at least 1 reader will be used. Each reader must access a shared counter value 250000000 times in the critical section. Note, it does not update anything, just "reads". For convenience code is below that will do this. A reader reads just one time and a writer writes just one time. Each reader needs to print its name when done. The writer will update the value 25000 times and print done. The writer will also set a shared flag, in-cs, when it enters the critical section and reset it just before it leaves the critical section. The reader must, upon entering the critical section, check this flag and write an error message if the flag is set. You can help us out for testing by using a version of the following code to give the writer a chance to run while readers are also running. So, start it in the midst of the readers.

```
k = (int) (numOfReaders/2);
for(i = 0; i < k; i++){
pthread_create(&readers[i], &attr[0], reader_thread, (void*) i);
}
/* Create the writer thread */
pthread_create(&writer[0], &attr[0], writer_thread, NULL);
for(i = k ; i < numOfReaders ; i++) {
pthread_create(&readers[i], &attr[0], reader_thread, (void*) i);
}
void relaxandspendtime()
{
int i;
for(i = 0; i < 250000000; i++) i=i;
}
```

You will again hand in the program on osnode16.cse.usf.edu by running /usr/local/os/turn_in4 and carefully following the instructions. Make sure your name is in the code file in comments!