Reader/Writer Locks

In this project, you are to:

a. Design and implement a readers/writers lock using semaphores that do not starve the readers and do not starve the writers;

b. Write the main C program that uses Reader/Writer locks;

c. Come up with a set of input scenarios that shows the behavior of your non-starving lock compared to the starving lock.

This is an **individual** project.

There are no required inputs to this program. You may use inputs if useful and explain use in the readme file. Output messages useful for testing and debugging.

- An input file, named `scenarios.txt,` that proves that your lock:
  - Is a correct readers/writers lock.
  - Does not starve the writers.
  - Does not starve the readers.
  - Each scenario takes one line (as the traces file in last project) ▪ An example and interpretation of this file is below:
    rwrrrrwrr wwrrrrwr
    This file contains two scenarios:
    1. One in which one reader arrives first, then a writer, then four more readers, another writer, then two more readers.
    2. And the second in which two writers arrive first, then four readers, one more writer, and one more reader.
  - You want these scenarios to test corner cases that are relevant for the point of your design: specifically, that writers will not starve. Thus, you design these test scenarios to make it possible for writers to starve. You do not need lots of readers/writers to make the case. You might want to limit each scenario to 10--15 readers and writers at the very most.

FILE ptr = fopen("scenarios.txt ","r");*