1. Drew Sadler
2. Grabs the last value from memory and adds a value to it, then returns it's new value to the pointer that was declared.
3. [asadler1@hopper3 Studio_12]$ ./atomic
   0
   10
4. [asadler1@hopper3 Studio_12]$ ./atomic
   starting value is 0
   final value is: 0
   Time elapsed is 1646415822 seconds
5. asadler1@hopper3 Studio_12]$ ./atomic
   starting value is 0
   final value is: 0
   Time elapsed is 1646417180 seconds
   [asadler1@hopper3 Studio_12]$ ./atomic
   starting value is 0
   final value is: 0
   Time elapsed is 1646417185 seconds
   [asadler1@hopper3 Studio_12]$ ./atomic
   starting value is 0
   final value is: 0
   Time elapsed is 1646417196 seconds

   Average time: 1646417187 seconds per increment

6. Average Time: 1646103292 seconds of Studio 11 per increment
7. Studio 12 performs the task 1.00019068973 times faster than studio 11
8. [asadler1@hopper3 Studio_11]$ ./mutex
   final value is: -14294381
   Time elapsed is 1646515303 seconds
   (comparing against average)
   98116 seconds longer without atomic instructions
9. I feel mutex locking would be better in situations where you want to make sure each time you are editing the variable one at a time, for instance any system you would want to have a locked system like editing a password or even accessing it, so that each time it is accessed or edited you know it either happened or was never performed . While for atomic instructions I feel it would be better in systems that require single operations to always finish, as in the sense of it can not be interfered with from other operations. So in systems like transferring data or messages through a thread, it would linearly do it one at a time till finish.