Group 3

Group Assignment 1

Question 1: The program outputs the ID of each thread accessing the do\_work function. These IDs can output in any order because each thread can access the do\_work function at a different time each time the program is run. Below are some, not all, possible outputs.

A close up of a sign

Description automatically generated

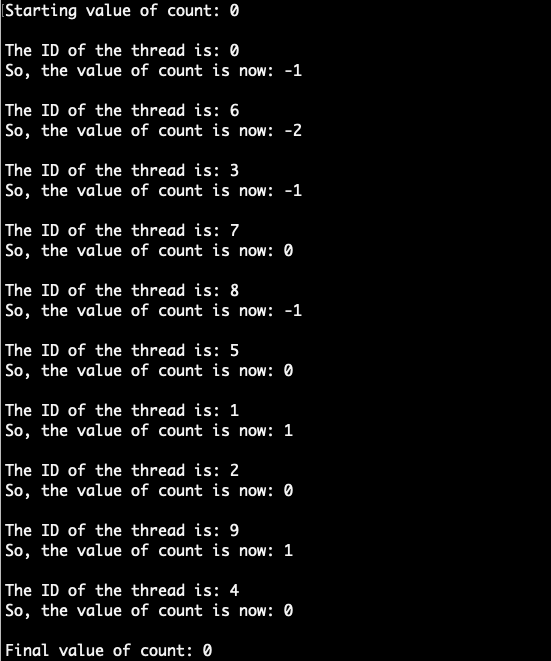
A close up of a sign

Description automatically generated

A close up of a sign

Description automatically generated

Question 2: Since there are the same amount of even numbered thread IDs as odd numbered thread IDs, the final value of the counter will be the same value that it was initialized at. This is because the counter gets incremented and decremented the same number of times. We initialized the counter to be zero, so the final print statement is zero. To show that the program is actually incrementing and decrementing the counter correctly, the code was temporarily edited with extra print statements to show the value of the counter after each thread accesses the do\_work function. Those results are below.



Question 3: The program checks which thread is in the function by if statements and then gives a lock to whichever thread gets to their critical section first. If thread one grabs the lock first, it decrements count and unlocks the lock. Thread zero can then grab the lock and proceed with incrementing count. If thread zero grabs the lock first, it must wait for thread one to decrement count before it can increment count. This is done with condition variables. Thread zero will wait and give up the lock so thread one can access its critical section and decrement count. Thread one will then signal thread zero to increment count. The two possibilities, thread one or thread zero going first, is shown in the screenshot of the output below.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated