

## Part 1:

I used a barrier because this way all the threads will complete the sum before the average can be calculated. My initial thought was to pull the average part out of the multithread but a barrier in this case does effectively the same thing. It manages concurrency in forcing the sum to be completed by each thread before it is read, prevents starvation by passing the number of threads to the barrier, and deadlock cannot occur because only one thread is using a shared resource which is not available until after the barrier.

```
mike@DESKTOP-0JMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A3P2$ ./a.out
Thread 0: Calculating sum from index 0 to 3.
Thread 0: Partial sum is 10.
Thread 1: Calculating sum from index 4 to 7.
Thread 1: Partial sum is 26.
Thread 2: Calculating sum from index 8 to 11.
Thread 2: Partial sum is 42.
Thread 3: Calculating sum from index 12 to 15.
Thread 3: Partial sum is 58.
Thread 4: Calculating sum from index 16 to 19.
Thread 4: Partial sum is 74.
Total sum is: 210, Average is: 10.50
```

## Part 2:

I used a mutex and conditional lock to complete the program. The mutex lock will make sure that only one player can hack at a time, and the conditional lock is only signaled for the thread that is the current active player. It is unlocked after the players turn and the conditional lock is removed after the current player is incremented. This achieves concurrency by only allowing one thread or player to be active at a time, prevents starvation by unlocking after the turn is complete and the player number is incremented. This ensures that the next player number is waiting and triggered to start after the current players turn is completed. Deadlock is prevented in the same way, by incrementing the player count before unlocking.

```
mike@DESKTOP-OJMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A3P2$ ./a.out
Player 1 is attempting to hack... ----- Current Player (1)
Player 1 succeeded in hacking!
Player 2 is attempting to hack... ----- Current Player (2)
Player 2 succeeded in hacking!
Player 3 is attempting to hack... ----- Current Player (3)
Player 3 failed to hack!
Player 1 is attempting to hack... ----- Current Player (1)
Player 1 succeeded in hacking!
Player 2 is attempting to hack... ----- Current Player (2)
Player 2 succeeded in hacking!
Player 3 is attempting to hack... ----- Current Player (3)
Player 3 failed to hack!
Player 1 is attempting to hack... ----- Current Player (1)
Player 1 failed to hack!
Player 2 is attempting to hack... ----- Current Player (2)
Player 2 succeeded in hacking!
Player 3 is attempting to hack... ----- Current Player (3)
Player 3 succeeded in hacking!
Player 1 is attempting to hack... ----- Current Player (1)
Player 1 failed to hack!
Player 3 is attempting to hack... ----- Current Player (2)
Player 3 failed to hack!
Player 2 is attempting to hack... ----- Current Player (3)
Player 2 failed to hack!
----- GAME OVER! -----

Player 2 wins with 3 points
```