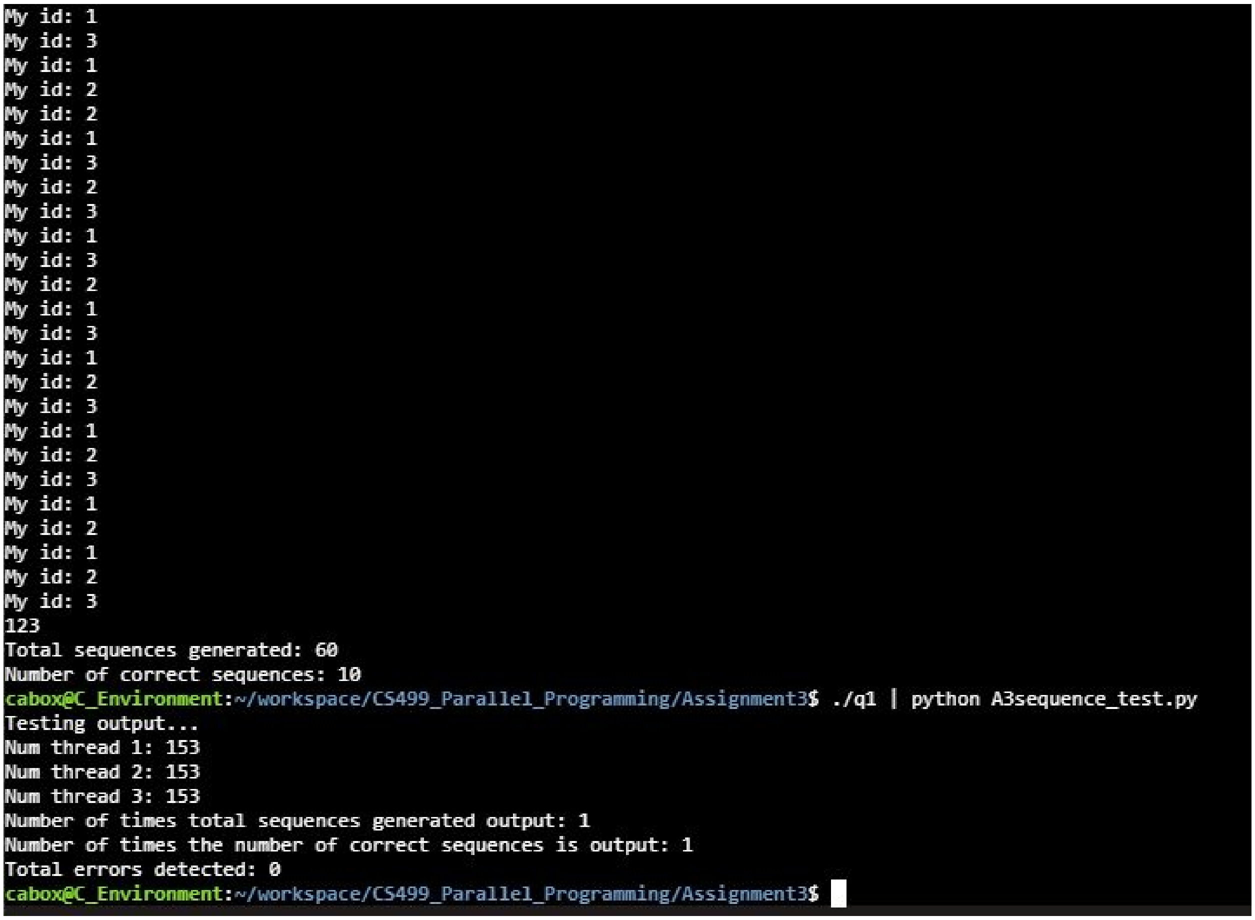
Brandon Horner
Michael Gowanlock
CS499: Parallel Programming
30 September 2019

Assignment 3: Programming with PThreads

Question 1)

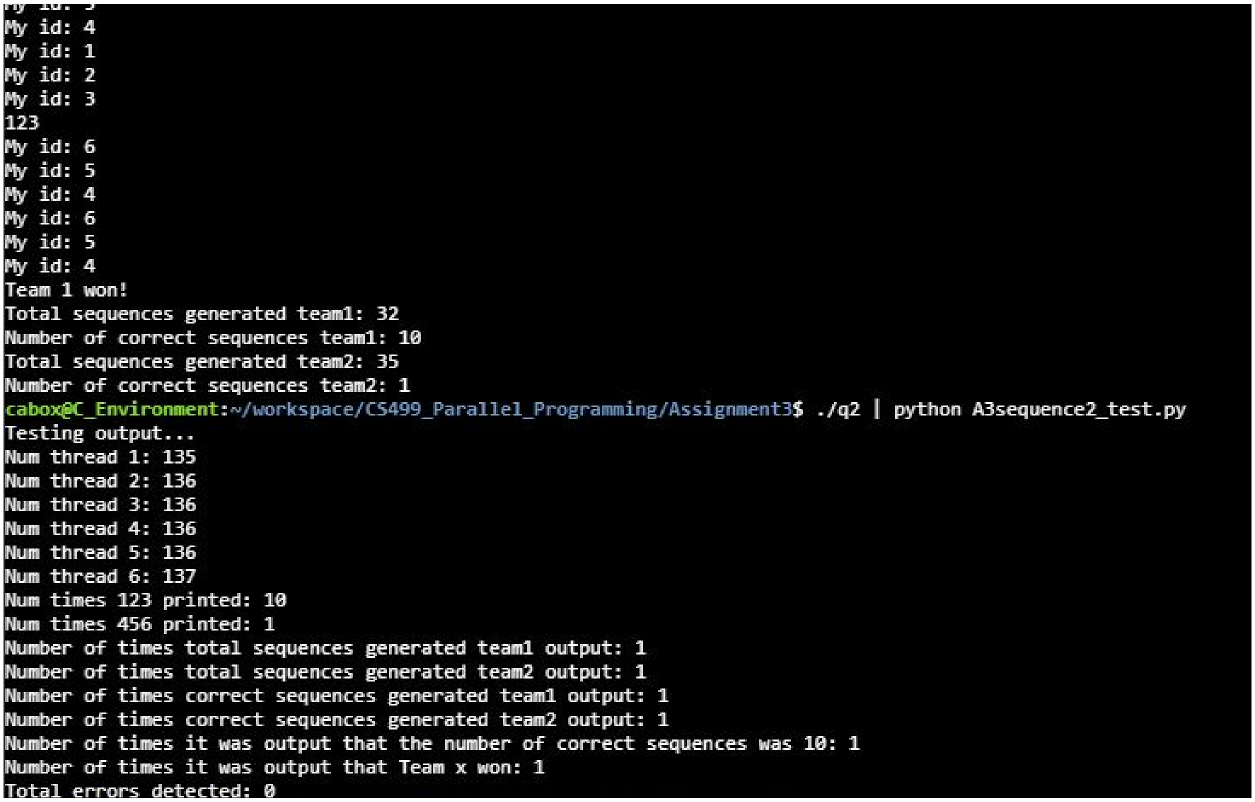
I started by utilizing the code in assignment 2 question 3. My first couple attempts at this problem did not work. I ended up needing a barrier to help synchronize the threads.



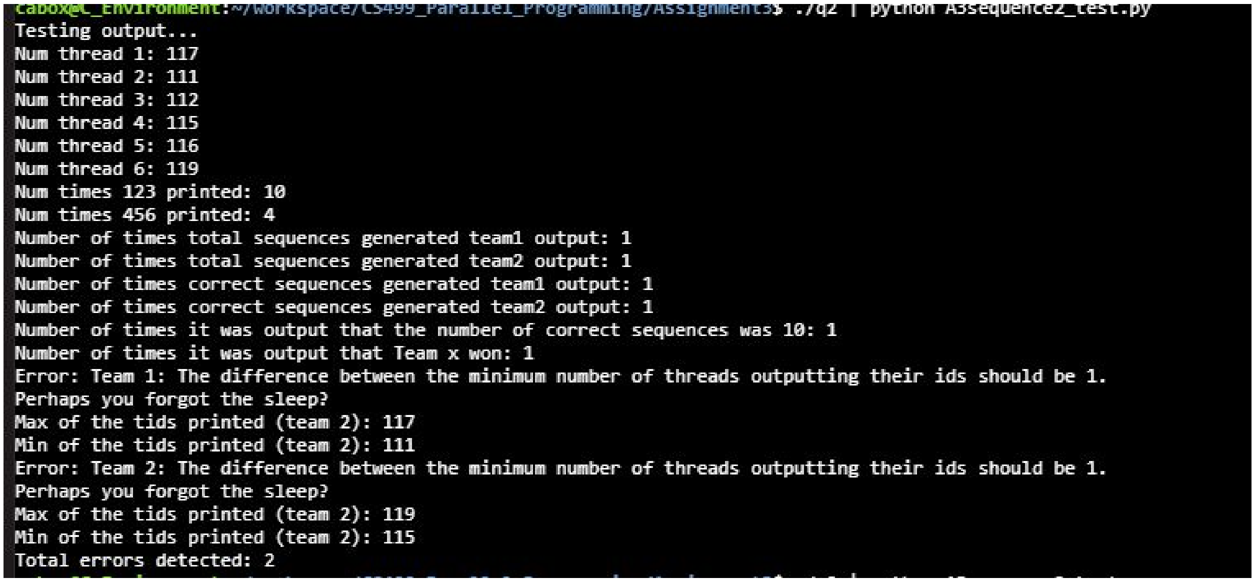
```
My id: 1
My id: 3
My id: 1
My id: 2
My id: 2
My id: 1
My id: 3
My id: 2
My id: 3
My id: 1
My id: 3
My id: 2
My id: 1
My id: 3
My id: 1
My id: 2
My id: 3
My id: 1
My id: 2
My id: 3
My id: 1
My id: 2
My id: 1
My id: 2
My id: 3
123
Total sequences generated: 60
Number of correct sequences: 10
cabox@C_Environment:~/workspace/CS499_Parallel_Programming/Assignment3$ ./q1 | python A3sequence_test.py
Testing output...
Num thread 1: 153
Num thread 2: 153
Num thread 3: 153
Number of times total sequences generated output: 1
Number of times the number of correct sequences is output: 1
Total errors detected: 0
cabox@C_Environment:~/workspace/CS499_Parallel_Programming/Assignment3$ 
```

Question 2)

I tried to copy paste the do_work() function from question 1 and slightly change it to modify a second set of variables and threads. Then when one team of threads finishes, it changes a variable to tell the other team to stop looping. However there seems to be a race condition somewhere. My code seems to stop half the time because one or more threads are waiting at a barrier indefinitely. Also, there seems to be a segfault occurring once in a while, however I could not find out why.

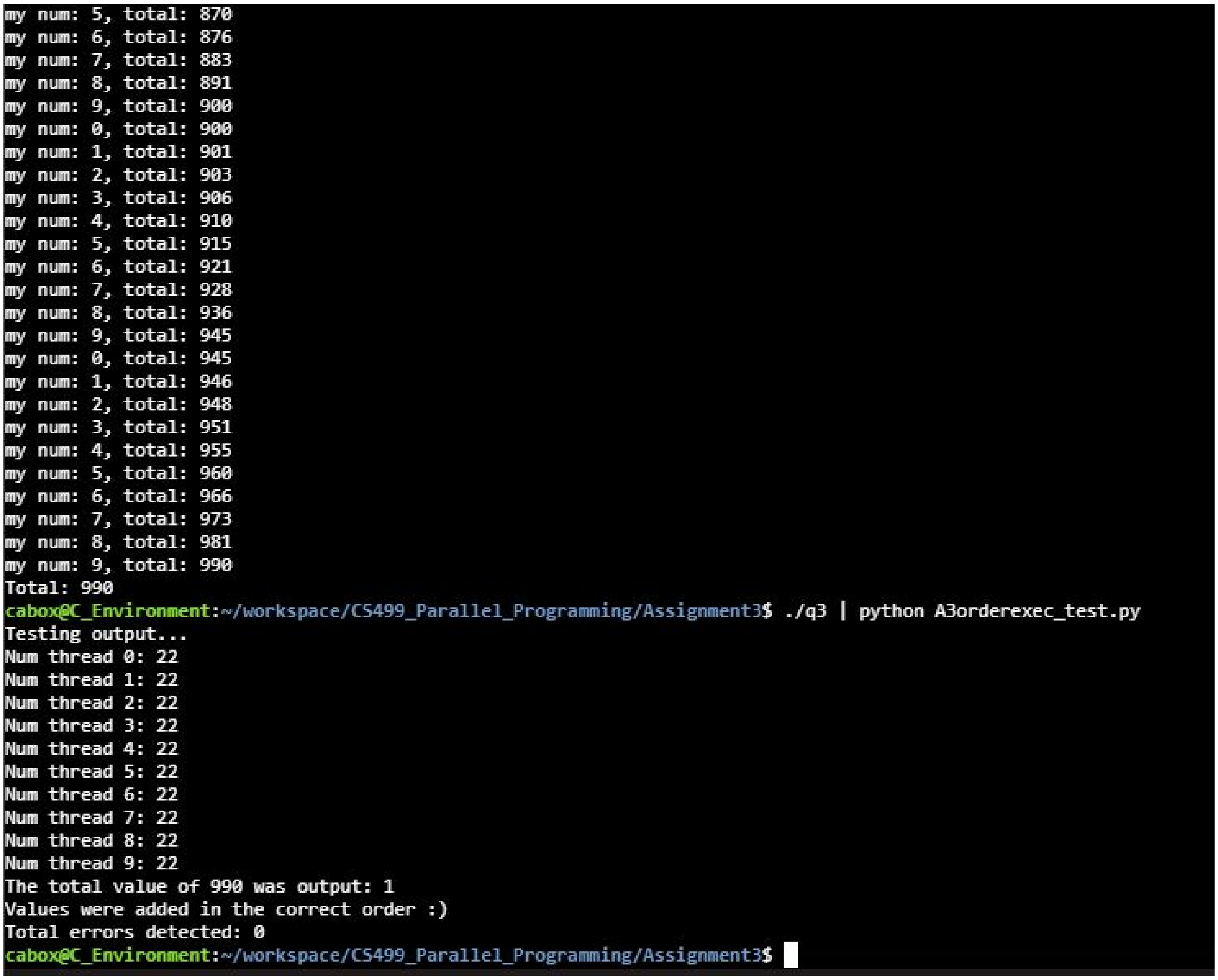


```
My id: 5
My id: 4
My id: 1
My id: 2
My id: 3
123
My id: 6
My id: 5
My id: 4
My id: 6
My id: 5
My id: 4
Team 1 won!
Total sequences generated team1: 32
Number of correct sequences team1: 10
Total sequences generated team2: 35
Number of correct sequences team2: 1
cabox@C_Environment:~/workspace/CS499_Parallel_Programming/Assignment3$ ./q2 | python A3sequence2_test.py
Testing output...
Num thread 1: 135
Num thread 2: 136
Num thread 3: 136
Num thread 4: 136
Num thread 5: 136
Num thread 6: 137
Num times 123 printed: 10
Num times 456 printed: 1
Number of times total sequences generated team1 output: 1
Number of times total sequences generated team2 output: 1
Number of times correct sequences generated team1 output: 1
Number of times correct sequences generated team2 output: 1
Number of times it was output that the number of correct sequences was 10: 1
Number of times it was output that Team x won: 1
Total errors detected: 0
```



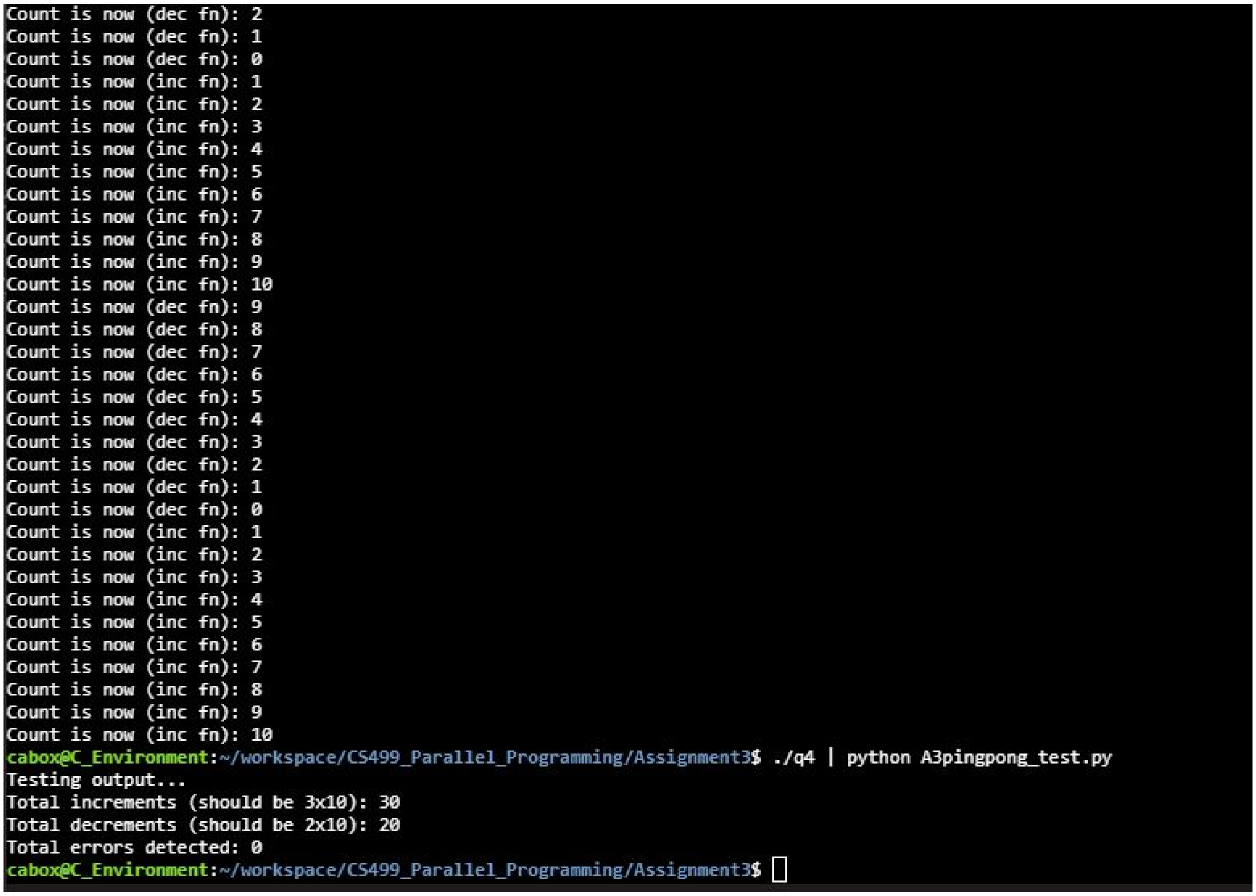
```
cabox@C_Environment:~/workspace/CS499_Parallel_Programming/Assignment3$ ./q2 | python A3sequence2_test.py
Testing output...
Num thread 1: 117
Num thread 2: 111
Num thread 3: 112
Num thread 4: 115
Num thread 5: 116
Num thread 6: 119
Num times 123 printed: 10
Num times 456 printed: 4
Number of times total sequences generated team1 output: 1
Number of times total sequences generated team2 output: 1
Number of times correct sequences generated team1 output: 1
Number of times correct sequences generated team2 output: 1
Number of times it was output that the number of correct sequences was 10: 1
Number of times it was output that Team x won: 1
Error: Team 1: The difference between the minimum number of threads outputting their ids should be 1.
Perhaps you forgot the sleep?
Max of the tids printed (team 2): 117
Min of the tids printed (team 2): 111
Error: Team 2: The difference between the minimum number of threads outputting their ids should be 1.
Perhaps you forgot the sleep?
Max of the tids printed (team 2): 119
Min of the tids printed (team 2): 115
Total errors detected: 2
```

Question 3) For this question, I almost tried many locks, but remembered that only one is necessary for one resource. When the thread is on the last thread, we reset the thread index to 0. I also have the thread return if it reaches 990 in count.



```
my num: 5, total: 870
my num: 6, total: 876
my num: 7, total: 883
my num: 8, total: 891
my num: 9, total: 900
my num: 0, total: 900
my num: 1, total: 901
my num: 2, total: 903
my num: 3, total: 906
my num: 4, total: 910
my num: 5, total: 915
my num: 6, total: 921
my num: 7, total: 928
my num: 8, total: 936
my num: 9, total: 945
my num: 0, total: 945
my num: 1, total: 946
my num: 2, total: 948
my num: 3, total: 951
my num: 4, total: 955
my num: 5, total: 960
my num: 6, total: 966
my num: 7, total: 973
my num: 8, total: 981
my num: 9, total: 990
Total: 990
cabox@C_Environment:~/workspace/CS499_Parallel_Programming/Assignment3$ ./q3 | python A3orderexec_test.py
Testing output...
Num thread 0: 22
Num thread 1: 22
Num thread 2: 22
Num thread 3: 22
Num thread 4: 22
Num thread 5: 22
Num thread 6: 22
Num thread 7: 22
Num thread 8: 22
Num thread 9: 22
The total value of 990 was output: 1
Values were added in the correct order :)
Total errors detected: 0
cabox@C_Environment:~/workspace/CS499_Parallel_Programming/Assignment3$
```

Question 4) I took most of my code from the increment/decrement slides on BBlearn. I just had to add a condition and signal to each threads calling function to signal to the other thread to start going. I also kept a total count because once we modified the count 50 times we were to stop the threads.



```
Count is now (dec fn): 2
Count is now (dec fn): 1
Count is now (dec fn): 0
Count is now (inc fn): 1
Count is now (inc fn): 2
Count is now (inc fn): 3
Count is now (inc fn): 4
Count is now (inc fn): 5
Count is now (inc fn): 6
Count is now (inc fn): 7
Count is now (inc fn): 8
Count is now (inc fn): 9
Count is now (inc fn): 10
Count is now (dec fn): 9
Count is now (dec fn): 8
Count is now (dec fn): 7
Count is now (dec fn): 6
Count is now (dec fn): 5
Count is now (dec fn): 4
Count is now (dec fn): 3
Count is now (dec fn): 2
Count is now (dec fn): 1
Count is now (dec fn): 0
Count is now (inc fn): 1
Count is now (inc fn): 2
Count is now (inc fn): 3
Count is now (inc fn): 4
Count is now (inc fn): 5
Count is now (inc fn): 6
Count is now (inc fn): 7
Count is now (inc fn): 8
Count is now (inc fn): 9
Count is now (inc fn): 10
cabox@C_Environment:~/workspace/CS499_Parallel_Programming/Assignment3$ ./q4 | python A3pingpong_test.py
Testing output...
Total increments (should be 3x10): 30
Total decrements (should be 2x10): 20
Total errors detected: 0
cabox@C_Environment:~/workspace/CS499_Parallel_Programming/Assignment3$ 
```