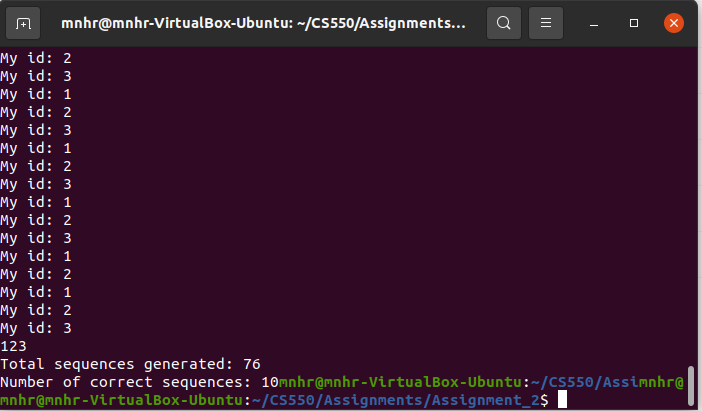
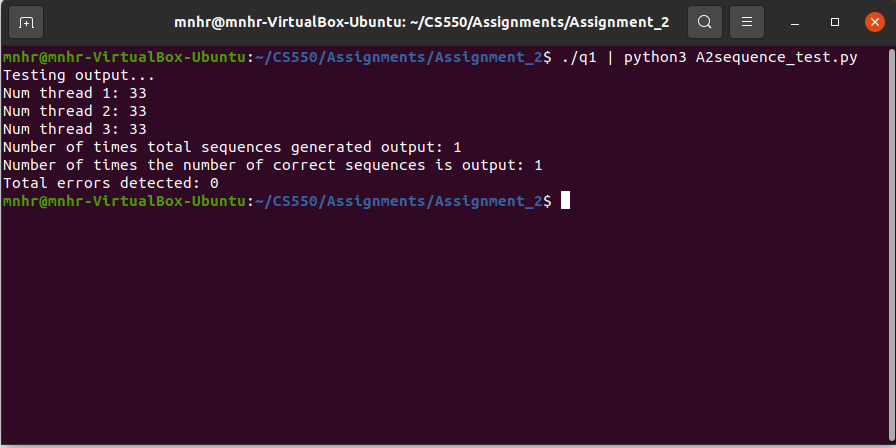
**Question-1:**

Screenshot of console output obtained from running solution of Question-1:



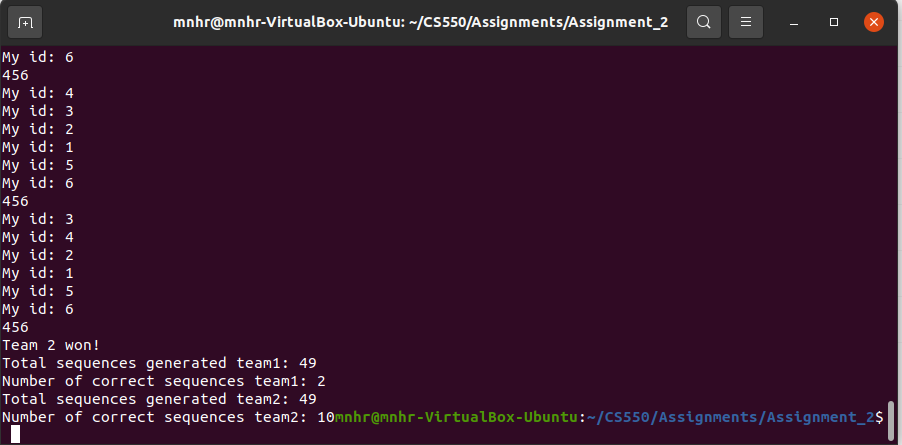
Output received by running with the test script:



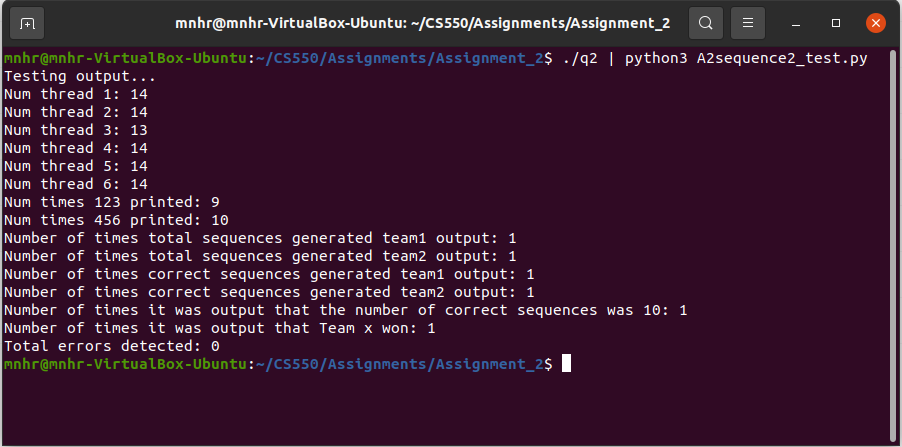
- As both of the screenshots display, the program works correctly. After generating the correct sequence “123” total 10 times, all three threads stop and return to the main thread. The main thread then prints the total number of sequences generated and the number of correct sequences generated.

**Question-2:**

Screenshot of console output obtained from running solution of Question-2:



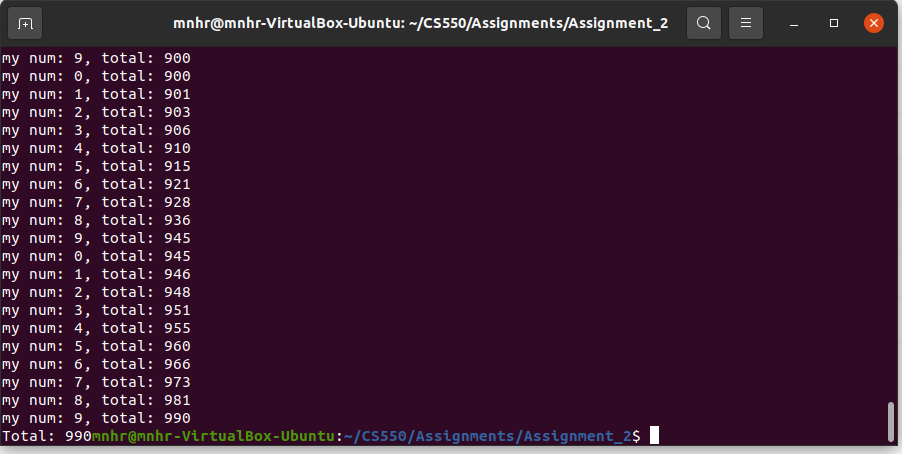
Output received by running with the test script:



- The program works correctly as displayed in both screenshots. The two sets of competing threads keep generating their sequences and one of the threads from the winner team updates the winner flag information which lets the threads from the defeated team know to stop generating sequences and return to the main thread.

**Question-3:**

Screenshot of console output obtained from running solution of Question-3:



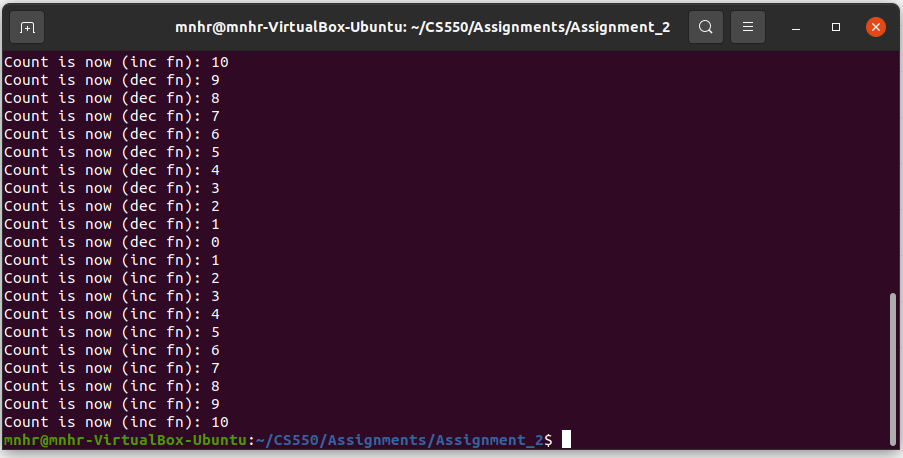
Output received by running with the test script:



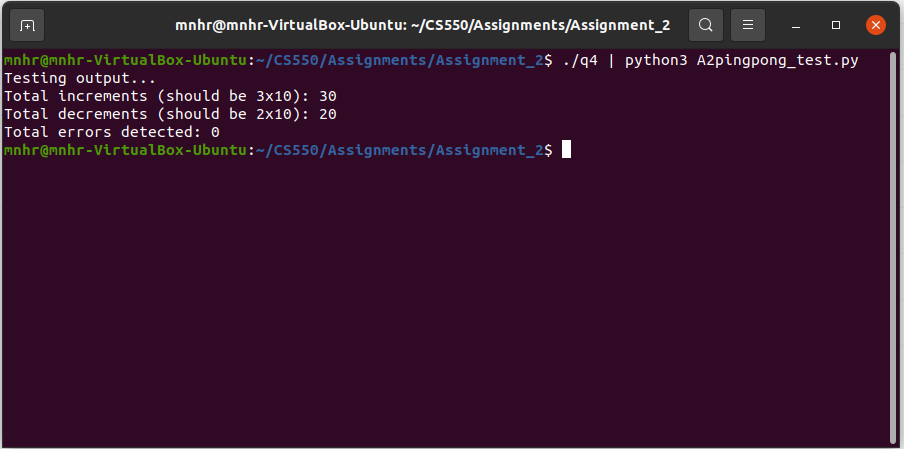
- The program works correctly. After each thread adds their respective values to the total in order 22 times which sums up to 990, they stop and return to the main thread.

**Question-4:**

Screenshot of console output obtained from running solution of Question-3:



Output received by running with the test script:



- This program works correctly. After the incrementer thread increments the counter value to 10, it signals the decrementer thread to proceed. After the decrementer thread decrements the counter value to zero (0), it signals the incrementer thread to start incrementing the counter again. Incrementer stops when it has incremented the counter value a total of 30 times and the decrementer stops when it has decremented the counter a total of 20 times.