For part 2 of this lab, we were assigned the task of creating a multi-threaded assignment that would calculate and share the values of a Fibonacci sequence into an array. The final output of the program is the display of the Fibonacci sequence(up to a specified number of values) and to also do the same for the sum of the sequence. This program does contain thread locking and as such the threads can be organized to complete first before executing final actions such as printing.

The CPU can move execution resources across threads in a multi-threaded process running on a single processor, resulting in concurrent execution. Thus making it an easier task to complete the process in order through different threads. This allows me as a developer to execute the sum command only after threads in charge of creating the Fibonacci sequence have finished setting its values.

**Make:**



**OUTPUT(testing for different values):**

