Due Date: April 1st

Lab 10: The Multithreaded Cook

In this lab, we'll practice multithreading. Using Semaphores for synchronization, implement a multithreaded cook that performs the following recipe, with each task being contained in a single Thread:

- 1. Task 1: Cut onions.
 - a. Waits for none.
 - b. Signals Task 4
- 2. Task 2: Mince meat.
 - a. Waits for none
 - b. Signals Task 4
- 3. Task 3: Slice aubergines.
 - a. Waits for none
 - b. Signals Task 6
- 4. Task 4: Make sauce.
 - a. Waits for Task 1, and 2
 - b. Signals Task 6
- 5. Task 5: Finished Bechamel.
 - a. Waits for none
 - b. Signals Task 7
- 6. Task 6: Layout the layers.
 - a. Waits for Task 3, and 4
 - b. Signals Task 7
- 7. Task 7: Put Bechamel and Cheese.
 - a. Waits for Task 5. and 6
 - b. Signals Task 9
- 8. Task 8: Turn on oven.
 - a. Waits for none
 - b. Signals Task 9
- 9. Task 9: Cook.
 - a. Waits for Task 7, and 8
 - b. Signals none

At the start of each task (once all Semaphores have been acquired), print out a string of the task you are starting, sleep for 2-11 seconds, then print out a string saying that you have completed the task. Finally, signal all the tasks that are waiting for you to finish. *Hint: You could store the semaphores in a 2D array.*

Grading Criteria:

 $Style/submission\ guidelines:\ \underline{https://gmierzwinski.github.io/bishops/cs321/style\ guidelines.html}$

Comments, Formatting, & Readability	5 Marks
Submission Guidelines	5 Marks
Program	30 Marks
Total	40 Marks