

Question 5

If a robot finishes a task and then is immediately assigned a new task, the robot may complete tasks forever. This means the robot will never put down the tools and the neighboring robots will never finish their tasks. In the dining philosopher problem this would be called starvation as some of the philosophers will never get to take a bite and therefore starve. A way to fix this would be to add a delay between a robot finishing its task and being assigned a new one. This would ensure that after a robot finishes its task, it will put the tools down so the neighboring robots can pick up the tools and finish their tasks.

Question 6

Some issues that arose during the development of the program

If all the actions take time to complete, then a race condition may occur. If there is space in the traffic pattern and one plane arrives and tries to enter the pattern but it takes a moment to enter. Then another plane arrives and sees that there is space in the traffic pattern before the first plane enters the pattern, there will be two planes trying to enter the pattern when there is only one spot. The same problem can happen if it takes time for the ATC to wake up. One Pilot may start to wake up the ATC and a second pilot may see the ATC is still asleep and try to establish a connection. Having time in between actions is a solvable issue thought, you need to use mutex, and keep variables from being modified by different threads at the same time. This was done in the program for #3 and is why it works so well 😊