

## Assignment 2 - Report

### Problem 1 – Sharing the bridge:

- For problem 1, I used repetition and large sample sizes to ensure my solution was reliable in preventing deadlock and starvation. Testing beyond NEON count 1000 for large groups of north and south farmers and checking over the print results showed that each farmer was getting a go at crossing the bridge. By having only one semaphore ensures that only one thread can enter the critical section at a time (mutual exclusion).

### Problem 2 – Share the bridge in pair:

- Problem 2 was harder to implement as there would be eventually a point when both north and south farmers would try to enter the critical section. At this point, as there is a semaphore with two locks, it could allow a north and south to enter. With my solution, I have multiple semaphores which wait for a partner and then when ready, move to the critical section and take both locks. By sleeping the south farmers in the beginning, allows the north farmers to start first but to sync and form a queue on the semaphore on their respective side which then prevents different farmers to enter the critical section until there is room.
- My semaphores release both locks only through one thread by using a counter to check when the 2<sup>nd</sup> of the pair has passed the point making sure another thread waiting doesn't enter the critical section without a pair.
- I tested with large sample sizes to find that the farmers use it in pairs and doesn't cause deadlock or starvation. If it is an odd number of farmers, they will wait forever and not prevent the other farmers from entering.

### Problem 3 – Multi-printer:

- Finally, Problem 3 also took some thinking to come up with a solution. Since monochrome and colour jobs cannot be printed at the same time, I created a Mode class which gave them their own queue and thread. Then a Printer class with three Job printer heads, which contains the critical section with the monitor (synchronised method) preventing deadlock. The modes will then try to use the printer and will be prevented by the monitor if it is in use ensuring mutual exclusion.
- Using the queues in the Mode, Jobs are selected by removing from the head making sure that the jobs are completed in order of entry.
- Also, by doing some simple maths, you can work out if more than one job can fit onto a head in a run of colour or monochrome mode. Therefore, reducing the time taken to print all the jobs on the queues.
- The output was the same as the description and when tested with a different input, the results were as calculated. It doesn't starve as it only finishes when the number of jobs is zero.