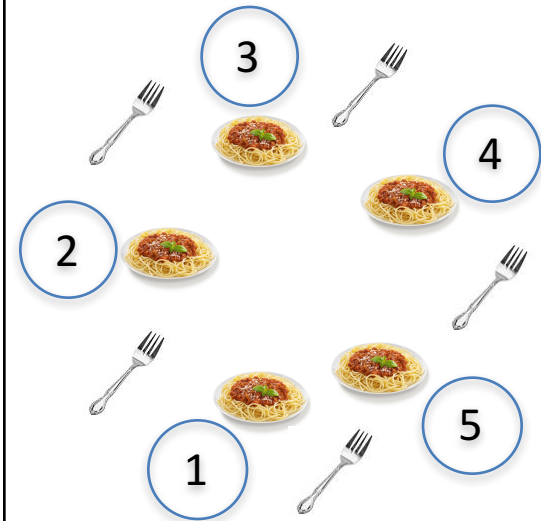


Homework: the dining philosophers problem

Dijkstra, 1965: Five silent philosophers sit at a round table with bowls of spaghetti. Forks are placed between each pair of adjacent philosophers. Each philosopher must alternately think and eat. However, a philosopher can only eat spaghetti when he has both left and right forks. Each fork can be held by only one philosopher and so a philosopher can use the fork only if it is not being used by another philosopher. After he finishes eating, he needs to put down both forks so they become available to others. A philosopher can take the fork on his right or the one on his left as they become available, **but cannot start eating before getting both** of them.

Eating is not limited by the remaining amounts of spaghetti or stomach space; an infinite supply and an infinite demand are assumed.



Deliver in 2 weeks (**Friday 09/09**): A **pdf** document with the description of the problem as you understand it and with the solution you choose to implement. Implement a solution for an unbounded number of philosophers, where each philosopher is implemented as a thread, and the forks are the synchronizations needed between them. Provide the **C source code** and a **makefile**, allowing for smooth compilation and execution.