1. Describe the asymmetric solution. How does the asymmetric solution guarantee the philosophers never enter a deadlocked state?

It ensures that neighbor nodes will go for the same resource so that a collision happens sooner, rather than letting every neighbor take one resource each.

2. Does the asymmetric solution prevent starvation? Explain.

Yes; a philosopher will be blocked on its first resource if a neighbor has already taken it as its second resource.

3. Describe the waiter's solution. How does the waiter's solution guarantee the philosophers never enter a deadlocked state?

With the waiter’s solution, a waiter mutex is locked while the philosopher takes the chopstick resources and when the philosopher releases the chopstick resources. If both resources are not available, a philosopher waits until they are available before using the waiter lock. Once finished, the neighbors will be signaled that the chopsticks are available.
This ensures that pairs of resources are allocated together and released together, so that no philosopher has just one resource and is stuck waiting.

4. Does the waiter's solution prevent starvation? Explain.

It doesn’t; a philosopher may be signaled that a chopstick on one side is available, but the neighbor at the other side might allocate its chopsticks during this time and a philosopher might be regularly stuck in-between two eating philosophers.

5. **Consider a scenario under a condition variable based solution where a philosopher determines at the time it frees its chopsticks that both chopsticks of another philosopher (Phil) it shares with are free, and so it sends the (possibly) waiting Phil a signal. Under what circumstances may Phil find that both of its chopsticks are NOT free when it checks?**

If, between the signal time and the pickup time, both chopsticks are taken by Phil's neighbors on both sides.