1. **Race Condition**

A race condition occurs when two or more threads can access shared data and they try to change it at the same time. Because the thread scheduling algorithm can swap between threads at any time, you don't know the order in which the threads will attempt to access the shared data. Therefore, the result of the change in data is dependent on the thread scheduling algorithm, i.e. both threads are "racing" to access/change the data.

1. **Disabling Interrupts**
2. Why is it impossible to achieve Mutual Exclusion via disabling interrupts on a multi-core machine?

Because nothing will change if you disable one, because the processes are always running on another core at the same time.

1. Why is it dangerous to give user processes the power to disable interrupts?

Disables multiprogramming even if another process isn’t interested in critical section.

1. **Peterson's Solution**
2. Play through the two scenarios of the handout of Peterson's solution. Document how it works.
   1. Process 0 and process 1 would be interested to set them true, after process 0 declares himself the loser, process 1 will do the same and making process 1 the only loser. Then process 0 can enter the critical region while process 1 wait. When process 0 is finished he sets interested on false an process 1 can enter the critical region.
   2. The Process 0 can enter the critical region without problems, but when Process 1 want to enter the region he has to wait until process 0 leaves the critical region and then Process 1 can enter.
3. Play through the scenario which makes the strict alternation approach fail. Document how it fails.

If you want to do two things, the one process want to do anything, you have to wait until the other process does something.

1. What is the meaning of the variable in Peterson's solution? When does it have any effect?

2 processes would enter the critical region nearly at the same time without this variable.

1. Extend the given functions such they can handle three processes.

I would expand the array to 3 slots and I would assign the process loser a number instead of the name loser.