CS241#20 – CSP II. Deadlock II and Dining Philosophers

1. Do we have a winner for the CRITICAL SECTION PROBLEM? Contestant #4:

```
Three shared variables: turn = 1, flagA = FALSE, flagB = False
```

thread1:	thread2:

flagA = TRUE
if(flagB) while(turn==2){ /* check again */}
// Do Critical Section stuff
turn = 2
flagA = FALSE
	flagB = TRUE
	if(flagA) while(turn==1){ /* check again */}
	// Do Critical Section stuff
	turn = 1
	flagB = FALSE

2. Deadlock

The _____ conditions for deadlock are:

_____: "A process is currently holding at least one resource and requesting additional resources which are being held by other processes."

_____:"There is a set of waiting processes, such that P_1 is waiting for a resource held by P_2, P_2 is waiting for a resource held by P_3 and so on until P_N is waiting for a resource held by P_1."

_____:"A resource can be released only voluntarily by the process holding it, after that process has completed its task"

_____:"At least one resource must be held in a non-shareable mode"

Three gardeners visit the garden shed pick up their desired tools for the day. There is a potential for deadlock. Fortunately they know about the C_____ conditions! Find four ways to solve the problem (break one condition each time). Name which condition you break in each case.

1

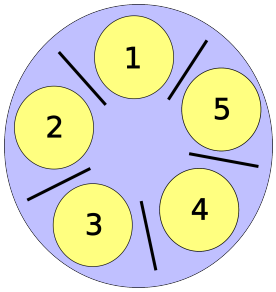
2

3

4

Remember Mergesort? How can you implement parallel Mergesort? Explain what synchronization calls you will use and when.

What is the Dining Philosophers problem?



Candidate Solutions:
1. "Pick up left chopstick. Pickup right chopstick. Eat. Release both."

2. "Pick up right. Pick up left. Eat. Release both"

3. "Eat when I tell you"

4. "Pick up left chopstick. Try to pickup right chopstick (Fail? release both and restart). Eat. Release both."

5?