Question 1. *(15 points) You are implementing a multilevel feedback queue (MLFQ) scheduler and are deciding what values to use for the time slice at each priority level. Your friend suggests simplifying the design by just using the same time slice value for each level. Is this a good or bad suggestion? Explain why – and be specific! You may find it useful to consider what would happen if you did use a uniform time slice value.*

Question 2. *(15 points) Explain whether user-level or kernel-level threads would be better suited to each of the following applications and why. Assume that you are running on a uniprocessor machine.*

1. *A weather modeling system in which large numbers of threads perform continuous computation.*
2. *A file sharing application in which each user is allocated a thread that reads and writes files stored on disk.*
3. *Suppose that you are running on a multiprocessor machine instead of a uniprocessor machine. Would your answer to parts 1 and/or 2 change? Why or why not?*

Question 3. *(15 points) Recall from class how we addressed the problem of busy waiting when implementing locks by introducing the idea of a 'guard' variable. Does this approach eliminate busy waiting? If it doesn't, explain why it preferable to the earlier, simpler approach where we implemented locks by calling test&set on the lock value itself.*

Question 4. *(20 points) Also from our final implementation of a test&set-based lock, the final step of a thread attempting to acquire a lock that's already held consists of setting the guard variable to zero and putting the thread to sleep. Explain why it is necessary to have these two operations occur atomically – i.e., what problem could occur if a context switch occurred after setting the guard to zero but before putting the thread sleep? Be specific.*

Question 5. *(15 points) In the producer/consumer problem, explain why the call to* `wait` *in the* `remove` *method must be wrapped in a* `while` *loop instead of a regular conditional.*

Question 6. *(20 points) Consider the following system state for four processes P_0 through P_3 and three resources A, B, and C in which we run Banker's algorithm to manage resource allocations.*

	Max			*Allocation*			*Available*		
	A	*B*	*C*	*A*	*B*	*C*	*A*	*B*	*C*
P_0	7	5	3	0	1	0			
P_1	3	2	2	3	0	2			
P_2	2	2	2	2	1	1			
P_3	4	3	3	0	0	2			
total				5	2	5	2	3	0

1. *Is the system in a safe state? Explain your answer.*
2. *Suppose processes P_0 requests new resources $(0, 2, 0)$. Will Banker's algorithm allocate these resources to P_0? Why or why not?*