

Matthew Browning
Dr. Li
Comp 2710
1 May 2020

Final Exam

- 1. Which of the following statements deletes the dynamic array?**
Answer is D
- 2. Which of the following statements related to pointers is incorrect?**
Answer is D
- 3. Which of the following statements is incorrect?**
Answer is B
- 4. Pointer variables are memory addresses and can be assigned to one another without regard to type.**
Answer is False
- 5. Recursive functions can be accomplished in one step, namely repeated calls to itself.**
Answer is False
- 6. A recursive function with parameter N counts up from any negative number to 0. An appropriate base case would be N == 0.**
Answer is True
- 7. A recursive function can have two base cases, as N == 0 returning 0, N == 1 returning 1.**
Answer is True
- 8. Revised solution 2 with three State variables regarding the Dining-Philosopher problem.**
 - 1. Please carefully review Solution 2 to list three states.**
The three states are THINKING, HUNGRY, and EATING
 - 2. How should the Semaphore elements of EatAgain be initialized?**
EatAgain should be initialized to 0
 - 3. How should the Semaphore mutex be initialized?**
Mutex should be initialized to 1
 - 4. What is the maximum number of Philosophers that can be waiting on a Semaphore element mayEat[i] at any given time?**
One philosopher is the maximum number
 - 5. What is the maximum number of Philosophers that can be waiting on mutex at any given time?**
Given that we have 5 philosophers, the maximum number of philosophers that we can have waiting is 4
 - 6. Does the code work correctly if the statement EatAgain[i].P() is moved before mutex.V() in take_chopsticks()? Briefly explain it.**
Moving this over prevents it from working because the philosopher waiting cannot reset the mutex, so no other philosopher could start waiting
 - 7. Does the code work correctly if the statements test((i+1)%5) and test((i+4)%5) are moved before state[i]=THINKING() in put_chopsticks()? Briefly explain it.**
The philosopher has to change its state before testing, so it will not work

9. In Fig. 0, suppose A and B are making simultaneous transfers between two accounts in a bank. Please predict potential threats for this transaction.

If a resource can only be released by the process holding it, after the process has completed its task, then A and B could not be released because each would have to be released for the other to complete their task

10. Please summarize the sources of major software developers' headaches from the concurrency mechanism. List at least 4 drawbacks.

There are a lot of headaches that can come from the concurrency mechanism, such as the conditions for a deadlock:

1. Only one process may use a resource at a time
2. A process may hold resources while waiting
3. A resource cannot be removed while a process holds it
4. A process holds at least one resource needed by the next process