Katie Prescott
Homework 2

1. If you run process 2 to completion, then A= $A + 2^{nd}\ row\ of\ C$=(02031), and process 2 gets marked. From here, no process has a row in R that is less than or equal to A. You cannot run process 1, because process 4 has all instances of resource 1. You cannot run process 3 because there are no more available instances of resource 5. Finally, you cannot run process 4 because process 1 has the only existing instance of resource 3. The deadlocked processes are 1, 3, and 4, since they are unmarked.

2. One way to eliminate the circular wait condition in the dining philosophers' problem is to number the forks, and make a requirement to grab the lower number fork of the two next to you, before grabbing the higher number fork. This will work because if you are between 1 and 2, you need to get 1 before you get 2, and it will cycle through, but if you are between 5 and 1, you have to grab the other fork first, since 5 is the highest number. In doing so, one philosopher will not be able to get a fork, and it will allow another philosopher to pick up 2 forks, therefore it will not deadlock.

3. If you start with 0, it will deadlock, because every process needs at least one more resource 3. If x is 1, then process D can run to completion. The available resources are now at 11221. Process A can run to completion, and the available resources are now at 21432. Then Process C runs, and the available resources are now at 32442. Then process B runs to completion and the available resources are now at 52552. The smallest value for x that makes this a safe state is 1.

4. Deadlock is not possible. If each process grabs one resource, then one of them will take the third resource, and run to completion, and release it's resources. Then the second process will grab one of the now available resources, and run to completion. No deadlocks will occur.

I pledge my honor that I have abided by the Stevens Honor System.