CS 4410: Operating Systems
Homework 3

Instructions for Homework 3:

- This is the third "k out of n" homeworks CS 4410.
- The homework may be done in pairs, or individually. If doing in pairs, one of you should upload to gradescope and add your partner to the group assignment in the upper right corner of the screen. (Do not just upload the assignment individually or it will be graded twice, which means grading will take longer.)
- The deadline is Thursday, October 20 at 11:59AM.
- No late submissions will be accepted.
- You must attribute every source used to complete this homework.
- Please add your solutions to **this pdf**, placing your answers in the boxes provided. This makes grading much simpler. *Thank you.*
- For two questions, you will need an integer, `Int1`, to be computed as follows:
  - If you are working with a partner, let `var` be the lexicographically smaller of the two NetIDs.
  - Let `varInt` be the integral part of `var`.
    That is, if `var = rst123`, then `varInt = 123`.
  - Let `Int1` be the first digit of `varInt`.
    That is, if `varInt = 123`, then `Int1 = 1`.

Student 1 Name: ______________________________, NetID: __________

Student 2 Name: ______________________________, NetID: __________

`Int1`: __________

1 Resource Allocation Graphs

In a recent SETI-at-home breakthrough, a new planet has been discovered with three-handed philosophers. There are two types of chopsticks: red and blue. Each philosopher needs 3 chopsticks to eat and all 3 chopsticks may not be the same color. Consider a table with `(Int1 mod 4) + 3` three-handed philosophers. There is one pile of `N` red chopsticks and one pile of `M` blue chopsticks on the table. Draw a Resource Allocation Graph that illustrates a deadlocked situation **with the largest possible value of `N`**.

Write down the number of philosophers calculated according to `Int1`.
Draw your RAG in the box below. Make sure to label all your nodes and edges.

`# of philosophers` = __________, `N` = __________, `M` = __________

2 Deadlock Detection Algorithm

As a reminder, the deadlock detection algorithm is:

Algorithm 1 Deadlock Detection Algorithm

1. free[] = avail[]
2. for all processes i: finish[i] = (alloc[i] == [0,0,...,0])
3. find process i such that finish[i] == 0 and req[i] $\leq$ free
4. if no such i exists, goto 8
5. free = free + alloc[i]
6. finish[i] = 1
7. goto 3
8. system is deadlocked iff finish[i] == 0 for some process i

1. Your setting for this question is the $(\texttt{Int1}+1)$th case in the list of cases obtained from `q2l1.txt`.

   Your Case No.: ____________
   Does a deadlock exist in the above system? **Your Answer:** ____________

   If yes, what are the `free` and `finish` arrays once the algorithm has progressed as far as it can?

   `free` = [____,____,____], `finish` = [____,____,____,____]

   If no, a possible execution sequence of these four processes is: ________________

2. Your setting for this question is the $(\texttt{Int1}+1)$th case in the list of cases obtained from `q2l2.txt`.

   Your Case No.: ____________
   Does a deadlock exist in the above system? **Your Answer:** ____________

   If yes, what are the `free` and `finish` arrays once the algorithm has progressed as far as it can?

   `free` = [____,____,____], `finish` = [____,____,____,____]

   If no, a possible execution sequence is: ________________

3 The Banker's Algorithm

3.1 Setup

Suppose you have a multi-process application in which each process requires some number of pages of memory. List the information needed in order to run the Banker's Algorithm.

Is it feasible for modern operating systems to utilize the Banker's Algorithm? Why or why not? (Limit your answer to 1-2 sentences.)

3.2 Assessing State 1

Process	Allocation	Max
	A B C	A B C
P0	1 4 1	5 6 2
P1	0 1 0	2 1 0
P2	1 1 1	5 6 1
P3	3 2 3	9 9 6
P4	2 1 1	3 2 1

Available
A B C
2 0 0

Is the above a safe state or an unsafe state? **Your Answer:** ______________

If it is safe, write the safe sequence here: ______________________________________

If it is unsafe:

- Which processes *are* able to finish (if any)? **Your Answer:** ______________
- When these processes complete what resources are Available?

  Your Answer: Available = [______,______,______]
- For each remaining process whose resource needs cannot be met, state the process, which particular resource it needs, how many units of that resource it still needs. For example:
  "PX still might need 5 more units of A, and only 4 are available."

- How could the state be made safe again? *Answer all that apply:*

  A Process 1 could finish without using more than its current allocation.
  B Process 2 could finish without using more than its current allocation.
  C Process 3 could request (0,1,2), be granted this request, and the resulting state would be safe.
  D Process 4 could request (1,0,1), be granted this request, and the resulting state would be safe.
  E It is possible for the state to be made safe again, but none of the above options accomplish this.
  F It is not possible for the state to be made safe again.

  Your Answer: __________________________

3.3 Assessing State 2

The following state differs from the one on the previous page only in P0's initial allocation.

Process	Allocation	Max
	A B C	A B C
P0	**3 2 2**	5 6 2
P1	0 1 0	2 1 0
P2	1 1 1	5 6 1
P3	3 2 3	9 9 6
P4	2 1 1	3 2 1

Available
A B C
2 0 0

Is the above a safe state or an unsafe state? **Your Answer:** ______________

If it is safe, write the safe sequence here: ____

If it is unsafe:

- Which processes *are* able to finish (if any)? **Your Answer:** ______________
- When these processes complete what resources are Available?

  Your Answer: Available = [_______,_______,_______]
- For each remaining process whose resource needs cannot be met, state the process, which particular resource it needs, how many units of that resource it still needs. For example:
  "PX still might need 5 more units of A, and only 4 are available."
- How could the state be made safe again? *Answer all that apply:*

  A Process 2 could finish without using more than its current allocation.
  B Process 3 could finish without using more than its current allocation.
  C Process 0 could request (1,1,1), be granted this request, and the resulting state would be safe.
  D Process 3 could request (1,0,0), be granted this request, and the resulting state would be safe.
  E It is possible for the state to be made safe again, but none of the above options accomplish this.
  F It is not possible for the state to be made safe again.

  Your Answer: ___________________________