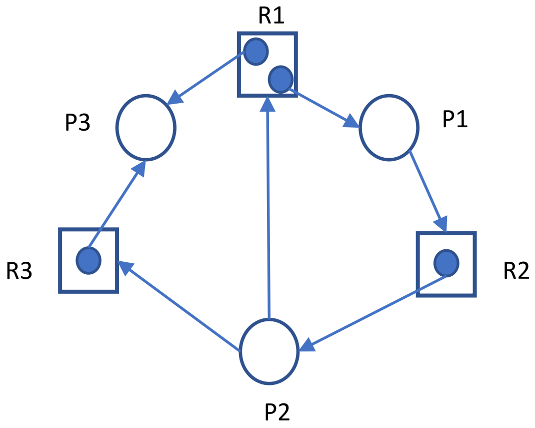
CSCE 4600
Spring 2020
Project #2
Due: April 30, 2020, per Canvas

For this project, you will be working in groups. Your task is to develop a system that can determine whether or not a particular resource allocation graph represents a deadlock state. A resource allocation graph, G(V, E), can be represented as an adjacency matrix M.



➔

X	P1	P2	P3	R1	R2	R3
P1	1	0	0	0	1	0
P2	0	1	0	1	0	1
P3	0	0	1	0	0	0
R1	1	0	1	1	0	0
R2	0	1	0	0	1	0
R3	0	0	1	0	0	1

Your program must read a text file which specifies the number of processes; the number of resources; the number of units of each type of resource; and the adjacency matrix M. Your program will determine if the state represented by M is a deadlock. **You may assume that the system's resource allocation policy is *expedient*!**

A text file containing the information from the example above has been posted to Canvas along with this project. Your program must be capable of reading text files which conform to the specification described within this file. Thus, your program may be graded using a different text file that conforms to this same specification.

You may choose to implement either one of the following deadlock detection mechanisms:

a. You may choose to implement the graph reduction algorithm to determine whether the graph represented by M is completely reducible and hence deadlock free;
b. You may determine that the state represented by G(V,E) or M is a deadlock state by inspecting the resource allocation graph for the existence of a knot.

Rubrics for this project are:

a. Description of your approach – 25pts
b. Implementation of your deadlock detection algorithm – 25pts
c. Correctness (your algorithm will be tested for different resource allocation graphs) - 25
d. Presentation and discussion of experimental results in the report – 25pts

Have FUN!!