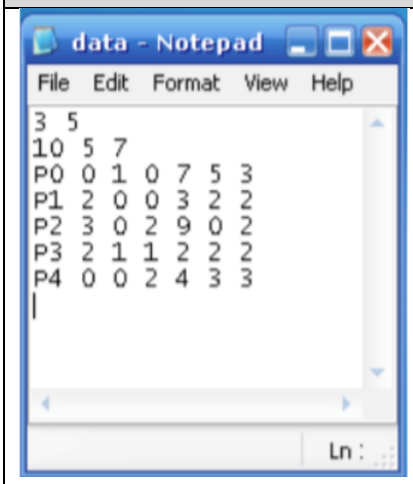
CMSC 125 Deadlocks – Banker's Algorithm – Safety Algorithm

GOAL	To be able to implement the Safety Algorithm of the Banker's Algorithm using C/C++ or Java programming language ONLY.
INPUT	The program must read from an input file. The input file contains the number of resource types, number of instances for each resource type, allocated resources for each process, maximum demand of each process. The sample text file below shows the format of these information. YOUR PROGRAM MUST INCLUDE, AS PARAMETER DURING INITIATION OF PROGRAM – NOT HARDCODED INTO THE SOURCE CODE, THE FILENAME OF THE DATA. IF YOU HARDCODE THE FILENAME INTO THE SOURCE CODE, YOU WILL AUTOMATICALLY GET ZERO POINTS FOR THIS TAKE HOME LAB EXERCISE. Important Note : Any source code copied from the internet will be DEALT WITH ACCORDINGLY. Think twice before submitting a requirement taken off the internet.
OUTPUT	The program MUST inform the user if the processes are in a deadlocked state or not. If not, the program must show the safe sequence of the processes.
Requirement during runtime	The program must be able to show the progress of determining if the processes may or may not enter into a deadlocked state. The program must also be able to show : 1. The Need matrix 2. The Available or Work matrix. (shown every iteration until the sequence is completed)

The Input file	
	• The 1st **line** tells that there are 3 resources types and 5 processes • The 2nd **line** gives the number of resource instances for each resource type: 10 instances for resource type 1, 5 instances for resource type 2 and 7 instances for resource type 3

- The 3rd to the 5th **line** give the information of each process
  - The **1st column** is the process name (P0, P1, etc)
  - The **2nd to the 4th column** shows the allocated number instances for each resource type
    - Example for process P0
      - 2nd column – 0 instance of resource type 1
      - 3rd column – 1 instance of resource type 2
      - 4th column – 0 instance of resource type 3
  - The **5th to the 7th column** shows the maximum number of instances for each resource type
    - Example for process P0
      - 5th column – 7 instances of resource type 1
      - 6th column – 5 instances of resource type 2
      - 7th column – 3 instances of resource type 3
    - Same interpretation goes for Processes P1 to P4

Refer to the sample output to the right using data.txt above and running it through the safety algorithm of the Banker's algorithm.

This is an example of a set of processes with a **safe sequence**.

Note : The example on the right ONLY SHOWS a safe sequence of processes. There are data that may result in an unsafe sequence. Your program must be able to handle that as well. When your code will be checked by the instructor, it will be tested on a file containing an unsafe sequence and another file with a safe sequence.

Banker's Algorithm - Safety Algorithm
Programmed by : Ritchie Mae T. Gamot

Reading data file...done...

The data file is interpreted below:

There are 3 resources with the following number of instances:

Resource 1(R1) = 10 instances
Resource 2(R2) = 5 instances
Resource 3(R3) = 7 instances

Allocated Resources for the 5 processes:

Process	R1	R2	R3
P0	0	1	0
P1	2	0	0
P2	3	0	2
P3	2	1	1
P4	0	0	2

Maximum Resources for the 5 processes:

Process	R1	R2	R3
P0	7	5	3
P1	3	2	2
P2	9	0	2
P3	2	2	2
P4	4	3	3

Need matrix for the 5 processes:

Process	R1	R2	R3
P0	7	4	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	4	3	1

Available instances for each resource type: (R1,R2,R3) = (3,3,2)

Running the Safety Algorithm...

P1 is given the resources...
Available is (3,3,2) – (1,2,2) = (2,1,0)
P1 is done with the resources...
Available is (2,1,0)+(3,2,2) = (5,3,2)

Current Safe Sequence is <P1>

P3 is given the resources...
Available is (5,3,2) – (0,1,1) = (5,2,1)
P3 is done with the resources...
Available is (5,2,1)+(2,2,2) = (7,4,3)

Current Safe Sequence is <P1,P3>

P4 is given the resources...
Available is (7,4,3) – (4,3,1) = (3,1,2)
P4 is done with the resources...
Available is (3,1,2)+(4,3,3) = (7,4,5)

Current Safe Sequence is <P1,P3,P4>

P2 is given the resources...
Available is (7,4,5) – (6,0,0) = (1,4,5)
P2 is done with the resources...
Available is (1,4,5)+(9,0,2) = (10,4,7)

Current Safe Sequence is <P1,P3,P4,P2>

P0 is given the resources...
Available is (10,4,7) – (7,4,3) = (3,0,4)
P0 is done with the resources...

	Available is (3,0,4)+(7,5,3) = (10,5,7) **Current Safe Sequence is <P1,P3,P4,P2,P0>**