The Bankers Algorithm

The bankers algorithm is designed to decide if a system is in a safe state, that is, without a deadlock based on circular wait. First, for the sake of the interface I used, it was necessary to read in the information from a text document, in which the amount of resources, the amount of each resource currently allocated to each process, the maximum amount of resources a process might need, and the current amount of each resource available.

After saving this information in the appropriate matrices, it is necessary to calculate each process need, which is the amount of each resource a process will have at maximum minus the amount that it currently has allocated.

Then, after properly collecting the data (which I have printed to the terminal screen for verification) the algorithm begins. It begins by finding the first process which has a need less than or equal to the amount of each resource available. (If it is ever found that no such process exists then the system is unsafe and the algorithm ends). It then takes the amount of resources that process had been allocated and adds it back into the amount of resources available. These steps are then repeated until the system is declared unsafe or a safe sequence of process completion is found. If such a sequence is found, it is printed out to the screen.

The Provided Problem

Considering a system with five processes P0 through P4 and three resources of type A, B, C. Resource type A has 10 instances, B has 5 instances and type C has 7 instances. Suppose at time t0 following snapshot of the system has been taken:

Snapshot at time T_0:

	Allocation	Max	Available
	A B C	A B C	A B C
P_0	0 1 0	7 5 3	3 2 2
P_1	2 0 0	3 3 2	
P_2	3 0 2	9 0 2	
P_3	2 2 1	2 2 2	
P_4	0 0 2	4 3 3	

Implement the Banker's algorithm to answer the following question: Is the system in a safe state? If Yes, then what is the safe sequence?

This is the data that is provided in input.txt, and when the algorithm is run we find that the system is in a safe state, and a possible safe sequence is:

P3 -> P1 -> P0 -> P2 -> P4

It is possible for there to be more than one safe sequence for the provided data that avoids deadlock.